Design • Build • Program • Learn • Compete

SERVO MAGAZINE

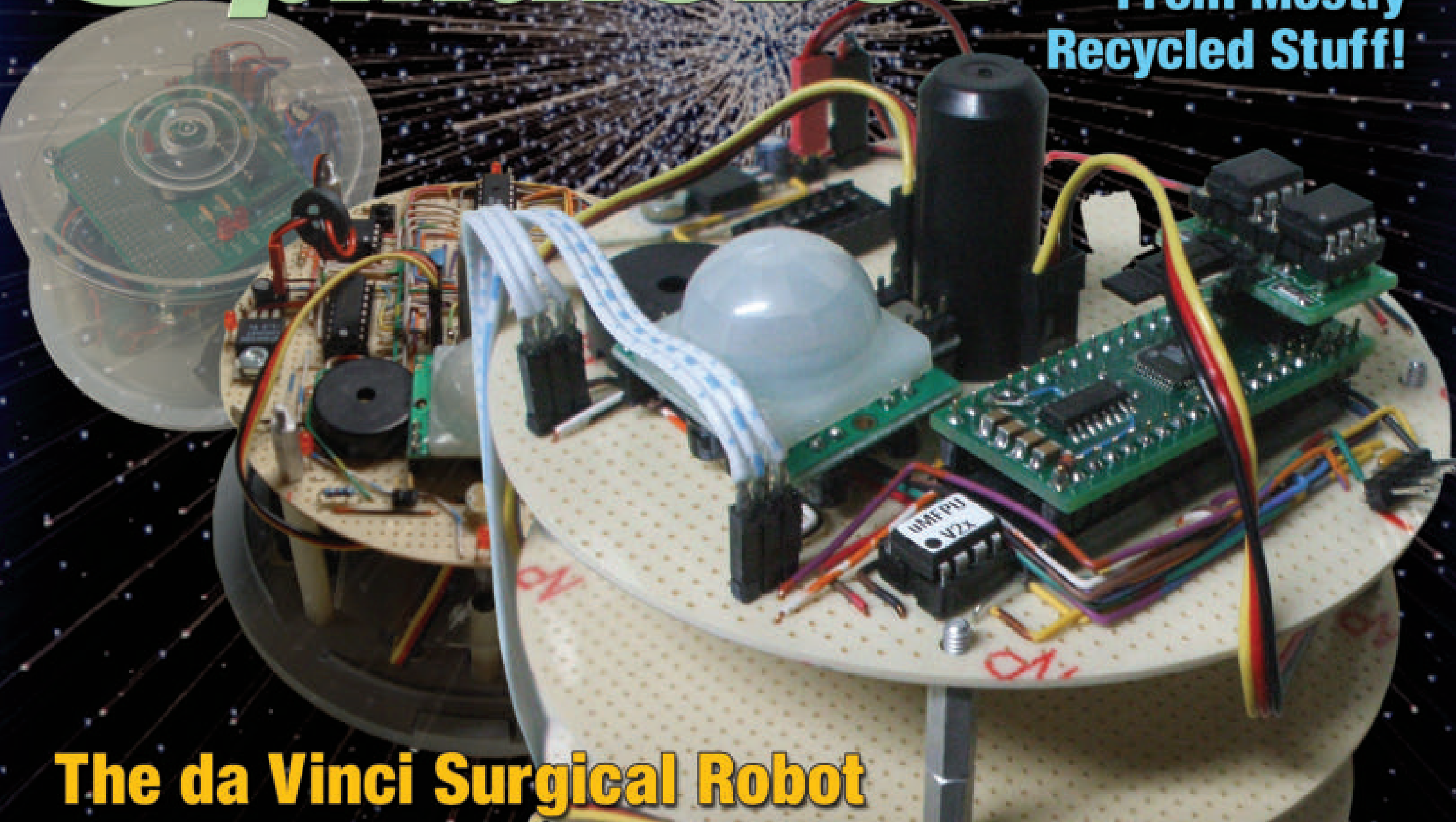
FOR THE ROBOT INNOVATOR

www.servomagazine.com

April 2012

Evolution Of The SpindleBot

A New Class Of Small Robot Easy To Build From Mostly Recycled Stuff!



The da Vinci Surgical Robot

An Inside Look

✦ Take the journey with *SERVO* author Tom Carroll, as he undergoes surgery via the da Vinci robot.

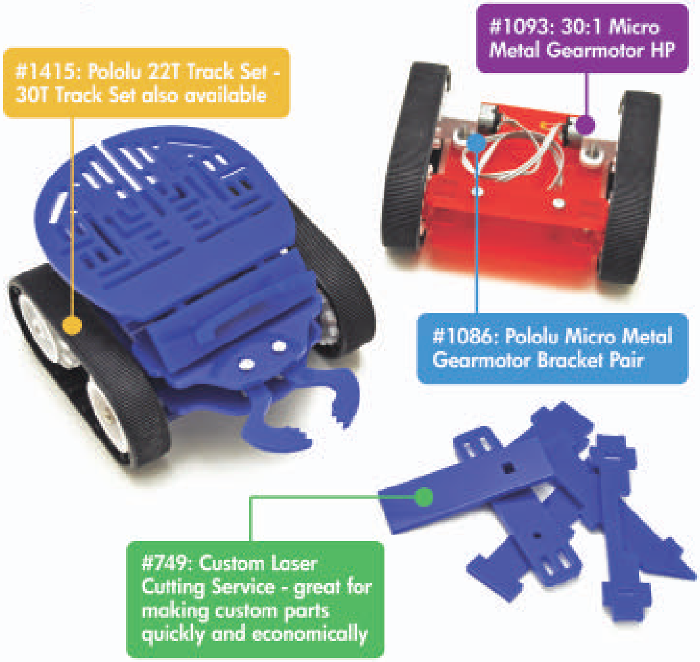
U.S. $5.50 CANADA $7.00

04

0 71486 02422 4

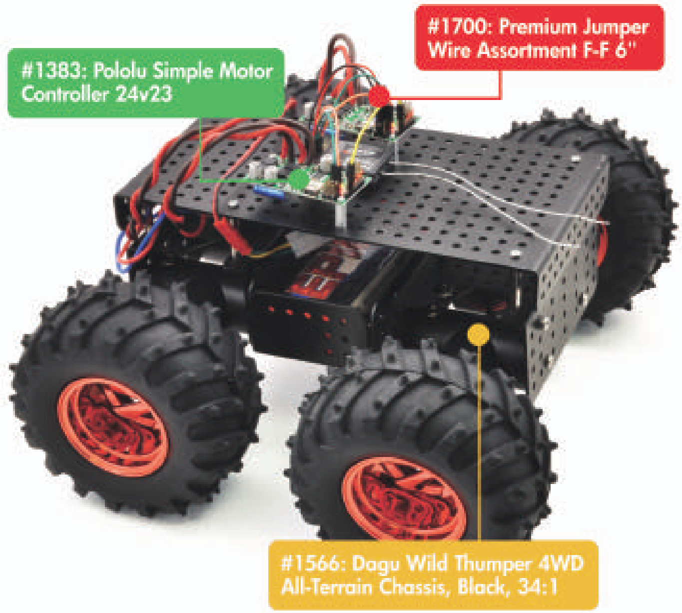
Custom Laser Cutting:

Design Your Own Chassis



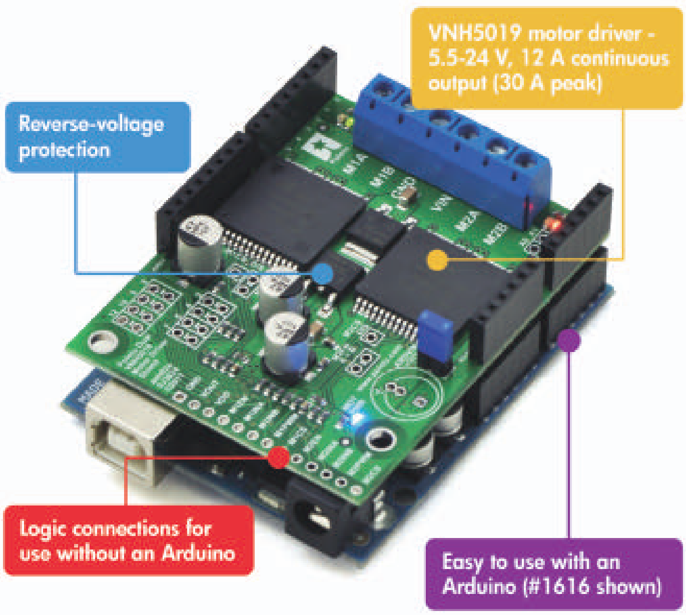
DIY Projects:

Wild Thumper-Based Robot



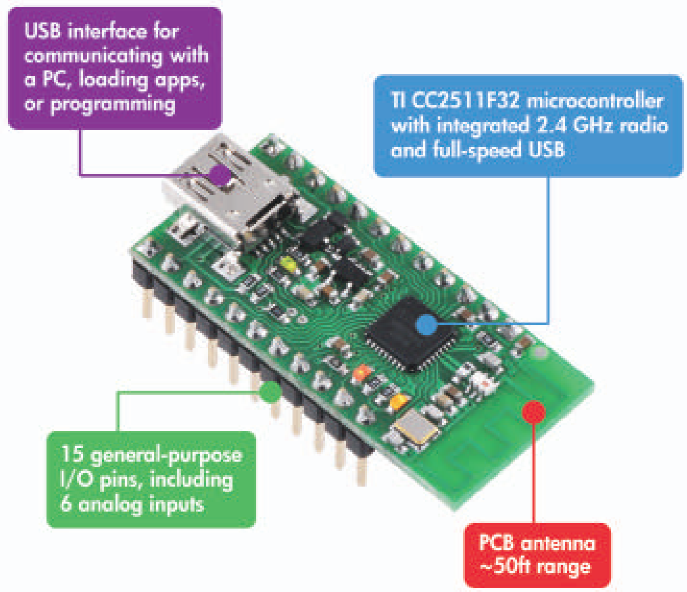
Motor Drivers:

Item #2502: Dual VNH5019 Motor Driver Shield for Arduino



Programmable Controllers:

Item #1336: Wixel Programmable USB Wireless Module



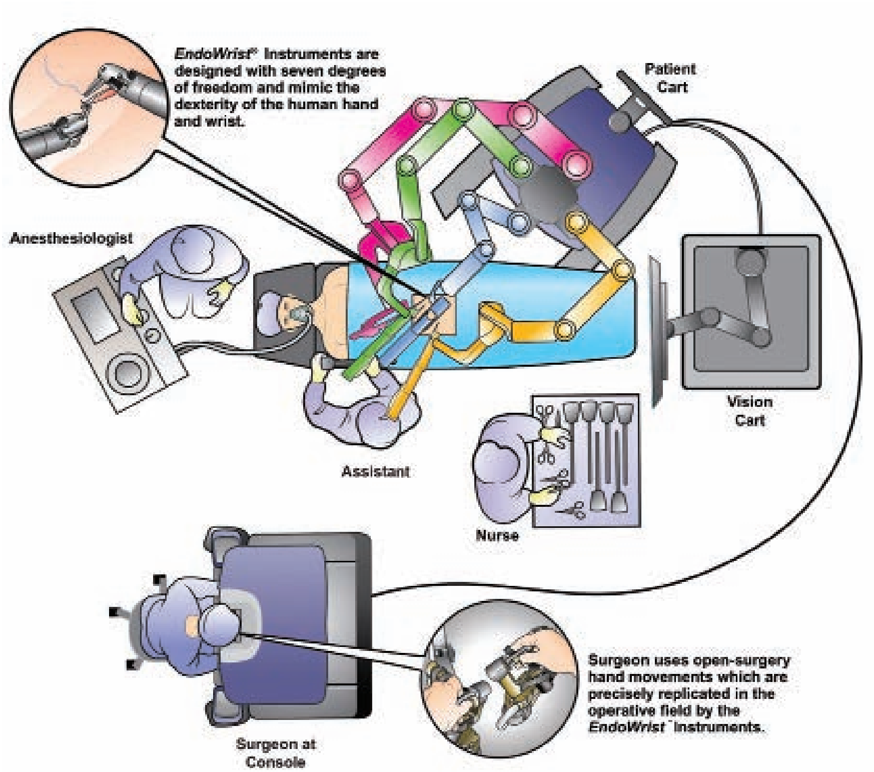
Finding the right parts for your robot can be difficult, but you also don't want to spend all your time reinventing the wheel (or motor controller). That's where we come in: Pololu has the unique products - from actuators to wireless modules - that can help you take your robot from idea to reality.

Pololu
Engage Your Brain

Find these products and more at **www.pololu.com**

SERVO MAGAZINE
04.2012
VOL. 10 NO. 4

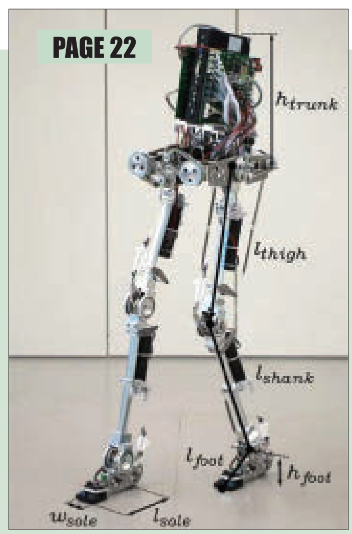
Columns



PAGE 74

PAGE 10

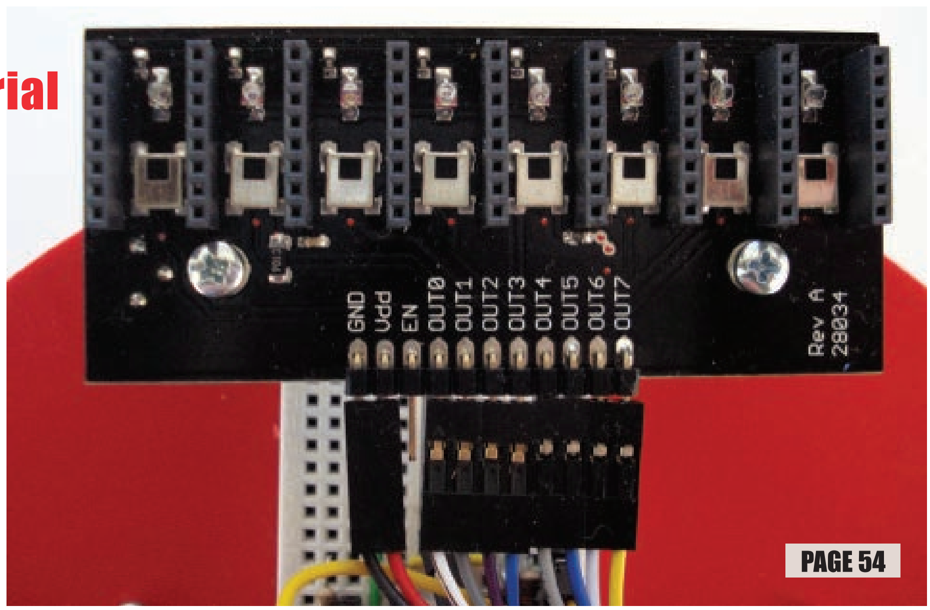
Departments



PAGE 22

SERVO Magazine **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@servomagazine.com

In This Issue ...



PAGE 54

The Combat Zone...

Mind / Iron

by Bryan Bergeron, Editor

Working With Light

A good friend asked if I could lend a hand developing a remote optical sensing device for his robot. The challenge was that communications between the remote environment and the local electronic sensors had to be carried via a single pair of optical fibers over several meters. Intrigued, I quickly agreed. After all — I asked myself — how hard could it be? I've worked with IR rangefinders, light radars, and even a pair of high voltage HeNe lasers. Plus, I had just finished experimenting with the IR LIDAR from a Neato vacuum teardown.

Well, the project that I assumed could be handled in a weekend has now stretched to over two months. It's one thing to work with off-the-shelf IR communications components and cables, but quite another to design an optical sensing system from the ground up. For example, even though most of my audio gear is connected via TOSLink optical cable, I had never tried to cut or splice the cable, or alter the end-cap lenses. I discovered that such an operation is non-trivial.

A challenge in working with fiber optics in general is the need for an infrastructure significantly different from that found in an electronics shop. Lacking the equivalent of a DMM or oscilloscope, I was virtually testing for battery voltage by placing the terminals on my tongue. That is, I was initially limited to using my visual perception of light intensity and area of illumination in place of quantitative tools.

I quickly learned that my drawers of standard electronic tools were useless when working with glass fiber. Not only is it fragile, but snapped-off ends have a tendency to seek out corneas like a heat-seeking missile. And once you break glass fiber, you might as well toss the entire fiber. That is, unless you have a $300 repair kit and know how to use it.

After wasting several hours polishing a few glass ends – the final step in a repair – I moved to plastic fiber. Plastic fiber is generally less efficient at carrying light than glass fiber, but is easier to work with. You can cut it with a sharp blade and then fire-treat the end to smooth it and reduce internal reflection. Of course, it helps to have a good pocket or even desktop microscope to examine the ends.

I love working with lasers, and this project was an excuse to purchase some hefty 100 mW IR red, green, and blue laser diodes; 100 mW lasers are "scary bright" – at least to me. Actually, 100 mW IR lasers are just plain scary. So, before powering up any of the laser diodes, I went online and ordered two pairs of laser goggles. It took two pairs to cover the four wavelengths of the lasers in my set.

As with glass and plastic fibers, laser diodes, and phototransistors, goggles are designed to work at specific wavelengths. Don't even think of using a pair of generic sunglasses for protection if you experiment with lasers of any significant intensity.

Although I found cheap no-name goggles on the Web that cater to the laser light show industry, I didn't trust the intensity reduction figures. Instead, I went to Thorlabs (**www.thorlabs.com**) for certified goggles. You might gasp at spending $150 or more for a pair of plastic goggles, but you'd probably gasp even more if you burned out your retinas because you trusted a $10 pair of no-name goggles. Another source for quality certified goggles is Phillips Safety Products (**www.phillips-safety.com**).

A major infrastructure technology is the perfboard equivalent in optical work: the optical bench. Think of a typical perfboard on steroids, with regularly spaced holes about 1/4" in diameter. Assuming you own standard

FOR THE ROBOT INNOVATOR
SERVO MAGAZINE

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct., Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore Only **1-800-783-4624**
www.servomagazine.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277, N. Hollywood, CA 91615

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
display@servomagazine.com

EDITOR
Bryan Bergeron
techedit-servo@yahoo.com

CONTRIBUTING EDITORS
Jeff Eckert, Jenn Eckert, Tom Carroll, David Geer, Dennis Clark, R. Steven Rainwater, Kevin Berry, Gordon McComb, Michael Ferguson, Robert Doerr, Mike Jeffries, Morgan Berry, Greg Intermaggio

CIRCULATION DIRECTOR
Tracy Kerley
subscribe@servomagazine.com

MARKETING COORDINATOR WEBSTORE
Brian Kirkpatrick
sales@servomagazine.com

WEB CONTENT
Michael Kaudze
website@servomagazine.com

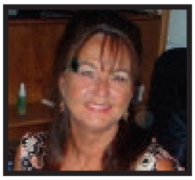
ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

PRODUCTION/GRAPHICS
Shannon Christensen
Sean Lemieux

mounts for your lasers, targets, lenses, mirrors, and other components, an optical bench makes setup quick and painless. I picked up a used 2 x 3 foot table-top optical bench on eBay. This isn't a must-have, but it beat mounting modules on mounds of putty or magnets tie-wrapped to the laser cylinders.

If you work with optics and lasers, you should know about Edmond Optics (**www.EdmundOptics.com**). It's a one-stop source for mirrors, lenses, and anything else you probably need to work with optics. I spent $30 on a "singlet" lens to concentrate the 3 mm beam from my red laser to about 1 mm. I wouldn't even think about working with a concentrating lens without safety goggles and precautions to minimize accidental reflections. There's a reason most of the hardware designed for laser and optics work is finished in dull black. Cover exposed stainless steel hardware with tape if you don't have time to paint it.

As demonstrated by the latest generation of robotic vacuum cleaners, semi-autonomous cars, and, of course, the military drones, optics and optical sensing are definitely part of the future of robotics. With a modest investment in infrastructure and safety equipment, you can make optics a part of your future experiments, as well. **SV**

Robytes

Discuss this article in the *SERVO Magazine* forums at http://forum.servomagazine.com.

by Jeff and Jenn Eckert

From Tsunami to Sukiyaki

Japan's Dream Project aims to reclaim devastated Miyagi prefecture areas as farmland.

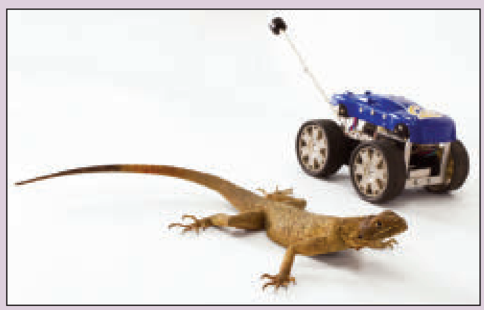
A pertinent question these days is, "What do you do when a huge chunk of your country is leveled by a 9.0 earthquake, swamped by a tsunami, and damaged by fallout from a nuclear plant meltdown, leaving fields inundated with salt, oil, and radiation?" If you're Japanese, the answer is, "Send in some robots." The country's Ministry of Agriculture, Forestry, and Fisheries (**www.maff.go.jp/e/index.html**, if you really want to know) recently announced that a program called the "Dream Project" will be implemented over the next six years in the Miyagi prefecture which was flooded by seawater and hit by fallout from the Fukushima plant a little over a year ago. The first step in the six-year project is to research how to bring a 600 acre site back to an arable condition. Long term, the ministry hopes to create a patch of farmland that is totally worked by unmanned tractors where CO_2 generated by the machinery will be channeled back into crops thereby reducing the reliance on fertilizers, and pesticides will somehow be replaced by LEDs that keep "rice, wheat, soybeans, fruit, and vegetables safe until robots can put them in boxes." The project will receive government funding of about four billion yen ($52 million), and high tech companies — especially bot builders like Panasonic — are being invited to jump in with another six billion yen of private investment capital. Land management during the six year period will be turned over to local farming corporations, and production will begin after they have figured out how to get rid of the salt. Once the project is up and running, local farmers will be encouraged to consolidate their own land under these corporations, eventually putting as much as 90+ square miles of once-fertile land back into production.

Telesmooching Arrives

Kissenger V. 1.0 provides long-distance kissing for desperate people.

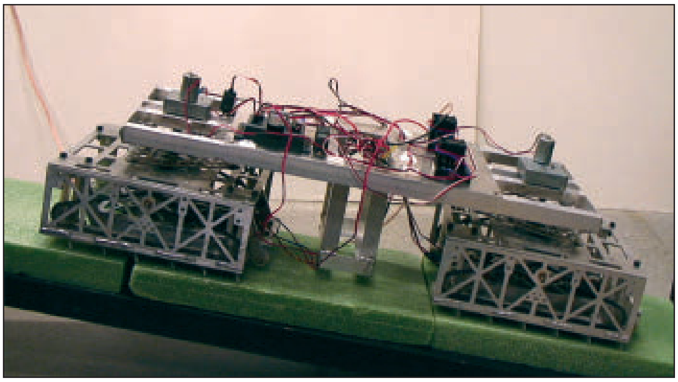
Moving from silliness to psychopathy, consider the latest product from Lovotics (**www.lovotics.com**) — a company dedicated to the research of human-to-robot relationships and "new possibilities for exploring the concept and possibilities of human love." No, it's not what you're thinking, but the company's Kissenger telepresence product is almost as creepy. The product consists of artificial lips attached to a pair of pig-like interactive devices that are supposed to provide "the convincing properties of the real kiss" with your partner over long distances. No price tag was cited, and it doesn't look like they are for sale online yet. No matter. I'm waiting until the version with a tongue is available. For a demo, see **www.youtube.com/watch?v=oSckuNlzQdM**.

Recommendation for Bots: Get a Little Tail



An Agama lizard provided inspiration for the UC Berkeley Tailbot.

Also inspired by the world of reptiles is a robotic car named Tailbot, developed at Berkeley's Mechanical Systems Control Lab (**msc.berkeley.edu**). It seems that Prof. Robert Full has spent the last 20 years researching how the toe hairs of geckos help them climb smooth, vertical surfaces and how their tails keep them from falling and allow them to right themselves in mid air. He recently tested a 40 year old theory that two-legged dinosaurs used their tails as stabilizers while running or evading predators. Not having any live dinosaurs on hand, some students used high speed videography to see how a red-headed African Agama lizard avoids spinning out of control when it leaps around. Sure enough, tails are pretty useful for controlling roll, pitch, and yaw (as any squirrel can tell you). The researchers proceeded to create a mathematical model of the lizard and Tailbot (a toy car equipped with a tail and a small position-sensing gyroscope). With the tail and no sensory feedback, Tailbot took a nose dive when driven off a ramp, but with feedback, it stabilized itself by using the tail to redirect its angular momentum. No practical applications of the concept were mentioned, but you have to figure that a dune buggy employing this form of "inertial assisted robotics" could be fun and pretty cool looking.



Georgia Tech's Scalybot 2 harnesses rectilinear motion.

Not So Serpentine Serpent

It sometimes appears that robot designers spend a great deal of time hanging out at serpentaria, and that includes Hamid Marvi — an engineering Ph.D. candidate at Georgia Tech (**www.gatech.edu**), creator of Scalybot 2. Snakebots are nothing new, but this one is a bit different. Snakes — it turns out — actually have four distinct ways of moving over land. There is the familiar serpentine movement in which they move along in a wavy fashion. But they can also travel by sidewinding (throwing their heads forward and dragging the rest of the body along), by employing the concertina method (bunching up and springing forward), and via the rectilinear method (a slow, straightforward creep). The latter is employed by the Scalybot — designed on the basis of a study of 20 species at Zoo Atlanta. As Marvi explained, "Snakes lift their ventral scales and pull themselves forward by sending a muscular traveling wave from head to tail. Rectilinear locomotion is very efficient and is especially useful for crawling within crevices, an invaluable benefit for search-and-rescue robots." The unit's reliance on rectilinear motion explains why it doesn't seem particularly snakelike in either appearance or motion. However, the machine — which is driven by four motors and controlled via a remote joystick — can change the angle of its scales when it encounters different terrains and slopes, adjusting them to either increase or decrease friction as required. A big advantage is that this type of motion doesn't use up much energy, so it should be able to travel over long distances and steep inclines without overheating or overtaxing its batteries. **SV**

GEERHEAD

by David Geer

Contact the author at geercom@windstream.net

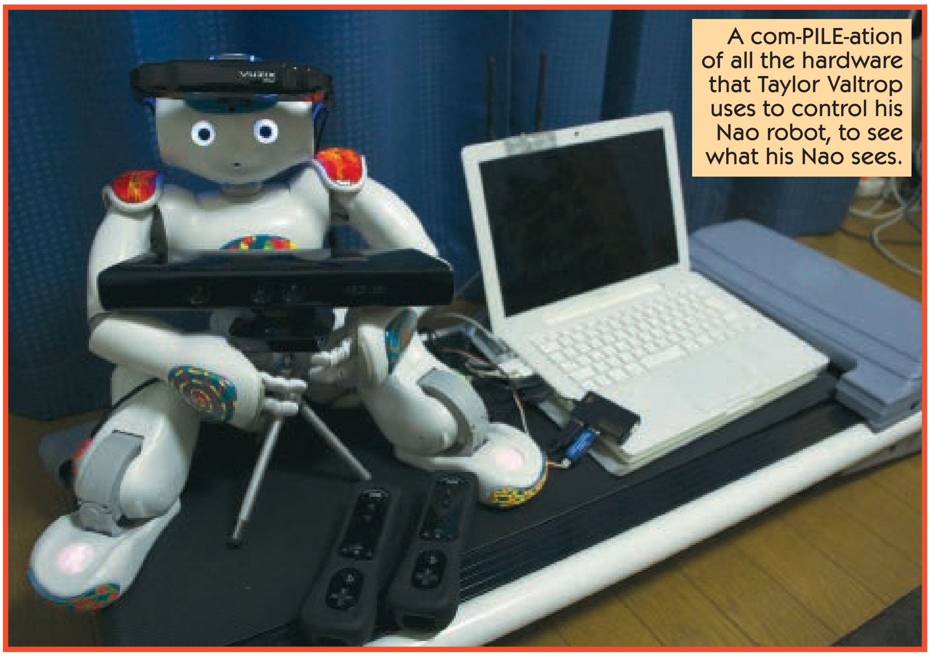
Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

Roboticist Achieves Human-Aided Robot Hand-Eye Coordination

Taylor Valtrop encountered some obstacles when he set out to program a Nao to exhibit autonomous hand-to-eye coordination. "I wanted the robot to autonomously interact with objects in its environment," Valtrop commented. Using home-built stereo cameras as a 3D sensor to retrieve the environmental data he needed, Valtrop tried to capture views that would enable the eye portion of the hand-to-eye interaction. However, the data quality was unsatisfactory.

Noting that the 3D data from a Kinect system was of much higher quality, Valtrop moved in that direction. "But the Kinect was too big to mount on my robot and it couldn't see short enough range to see the robot's own hands, so this failed to meet my needs as well," he explained.

Valtrop ultimately settled on using the Kinect system to collect his own skeleton position data so that he could feed that into the robot. In the end, Valtrop was able to achieve some remarkable human-aided, hand-to-eye coordination activities on his Nao robot.



A com-PILE-ation of all the hardware that Taylor Valtrop uses to control his Nao robot, to see what his Nao sees.

Teleoperation

There are many software libraries available for Kinect that make the software accessible. "I use OpenNI/NITE which is easily discovered using Google," Valtrop says. Valtrop was able to make the Kinect software, hardware, and the eyewear work for his purposes without any modifications. He simply used the available software libraries to access the data from these systems and fed that into his own system. The Nao is a stock robot with no customizations for the current research.

However, the data from Kinect is not instantaneously ready for use in robot control. "The pure data from the Kinect is a depth map which is a 2D image that gives the distance of each point in the image to the Kinect

camera," explains Valtrop. Taylor uses the NITE middleware (software) with OpenNI to retrieve the user's skeleton data from the 2D image. "This gives me positions and rotations of a user's joints. The problem is that this data is relative to the Kinect camera, so it does not directly translate to the joint angles that the robot should perform," Valtrop continued.

"So, for example, I might have my right upper arm bent at 20 degrees to the right of my torso, but my torso is rotated 10 degrees to the right relative to the Kinect camera. So, the Kinect tells me that my right upper arm is rotated 30 degrees to the right. Furthermore, I can ask the Kinect for the position of my right shoulder, elbow, and hand. From that data, I can use trigonometry to reduce the pure angle of the elbow joint to send to the robot," Valtrop commented.

Controlling the robot's walking is much simpler. "When I stand up (if my torso is above 0.5 meters), the robot executes a static stand up animation. When my body is rotated, the robot rotates. When my body is translated forward or backward, the robot walks in the corresponding direction. When I step to the left or right, the robot side steps also. If I side step a little, the robot goes slowly. If I side step a lot, the robot moves more quickly," illustrates Valtrop.

Valtrop uses a treadmill in his command and control arsenal to correct his position to make it so he is always standing in the same place. "I can keep walking forward and the treadmill tries to bring me back to the center position," says Valtrop. Unfortunately, the treadmill is simple and unidirectional so that Taylor must reset his position with a rotation or side step.

The HMD unit controls only the robot's head position. The Wii remotes open and close the robots hands and rotate its wrists. "I use this because I cannot use detailed hand recognition with the Kinect while I am using the complete skeleton detection," Valtrop explained.

Joint Control

Valtrop uses a couple of different approaches to control the Nao robot's joints using the Kinect hardware and software. "In both examples, linear algebra (matrix multiplications) is vital for calculating relative rotations between reference points to rectify the issues mentioned previously about all of the data coming in relative to the Kinect camera," says Valtrop.

In the first joint control approach, Valtrop looks at each joint's data as it comes from the Kinect and uses trigonometry to figure out what angle to send to each joint on the Nao robot. "It is a bit brute force, and it is not completely accurate, but it is fast and works in near real time. I can even dance in sync with the robot," Valtrop declares.

The reason this approach is not entirely accurate is because even if the Nao mimics the angles of Valtrop's own joints accurately, this can lead to the robot having its hand in a different position than his, due to the difference in proportions between the Nao's body and his. "So, it is great for dancing but not for manipulating tools," Valtrop reports.

In the second joint control approach — which is much more automated — Valtrop only needs to compare his hand's position relative to his torso and calculate the difference. "Then, I send that difference to Nao as a goal to its hand position. Nao then uses its built-in inverse kinematics engine to determine by itself the optimum position of each joint to accomplish the goal position,"

Taylor Valtrop with his Nao and his cat, Lotus.

A close-up of the laptop and cabling used in Valtrop's configuration.

Nao sees Lotus the cat so Valtrop also sees Lotus using his Vusix eyewear.

Valtrop's view of what Nao sees: Lotus getting comfortable.

Valtrop explains.

This method is a bit slower than the other one and dancing does not work in real time. However, it is very accurate for the hand motions and it feels more natural when the operator is manipulating tools in the hands of the robot.

Still, even though Nao's hand is in the same relative position as Taylor's, its elbow could be in a completely different pose, so it is not usable all of the time (in the case of obstacle avoidance, for example).

Intelligence, Simulation, and Application

Valtrop uses a Nao in simulation as he develops the software for his uses. This is a safety precaution. "The simulation is so that I can determine whether what I am doing will cause Nao to fall easily, and to check whether there is a problem with my math that would make the robot move in a damaging way," Valtrop asserts. Speaking of math, the algorithms in use are inverse kinematics (linear algebra), whatever PrimeSense (the maker of the closed source library for OpenNI's NITE middleware package) is doing for the skeleton data, and a lot of trigonometry. "I use a mix of C++, Python, and Nao's 'Choreograph' behavior editor GUI," he comments.

As for practical applications of the technology, Valtrop's favorite teleoperation is using the Kinect technology to do Ikebana (which is a form of Japanese flower arrangement). "My favorte Nao teleoperation is when I cut a banana and a long green onion with Nao using a knife," Valtrop asserts. In that same video, Valtrop also manipulates a chess board and removes a tissue from a box. "I tried to hammer a nail too, but that didn't work out so well," says Valtrop.

Valtrop has also tried to play miniature golf through the robot, as well as miniature pool/billiards. There is no video for those attempts. "I once watered my plants remotely, grabbing a watering pot and bringing it to the plants. But every time, I spilled so much water that the video didn't turn out well," Valtrop explains.

Finally, Valtrop had Nao use a Geiger counter to search his apartment for a radiation source. "The bananas I was using for this did not provide enough noticeable radioactivity to convince a viewing audience that I was actually successfully finding them instead of just directing Nao toward where I knew I had hidden them," Valtrop concluded.

Last Word

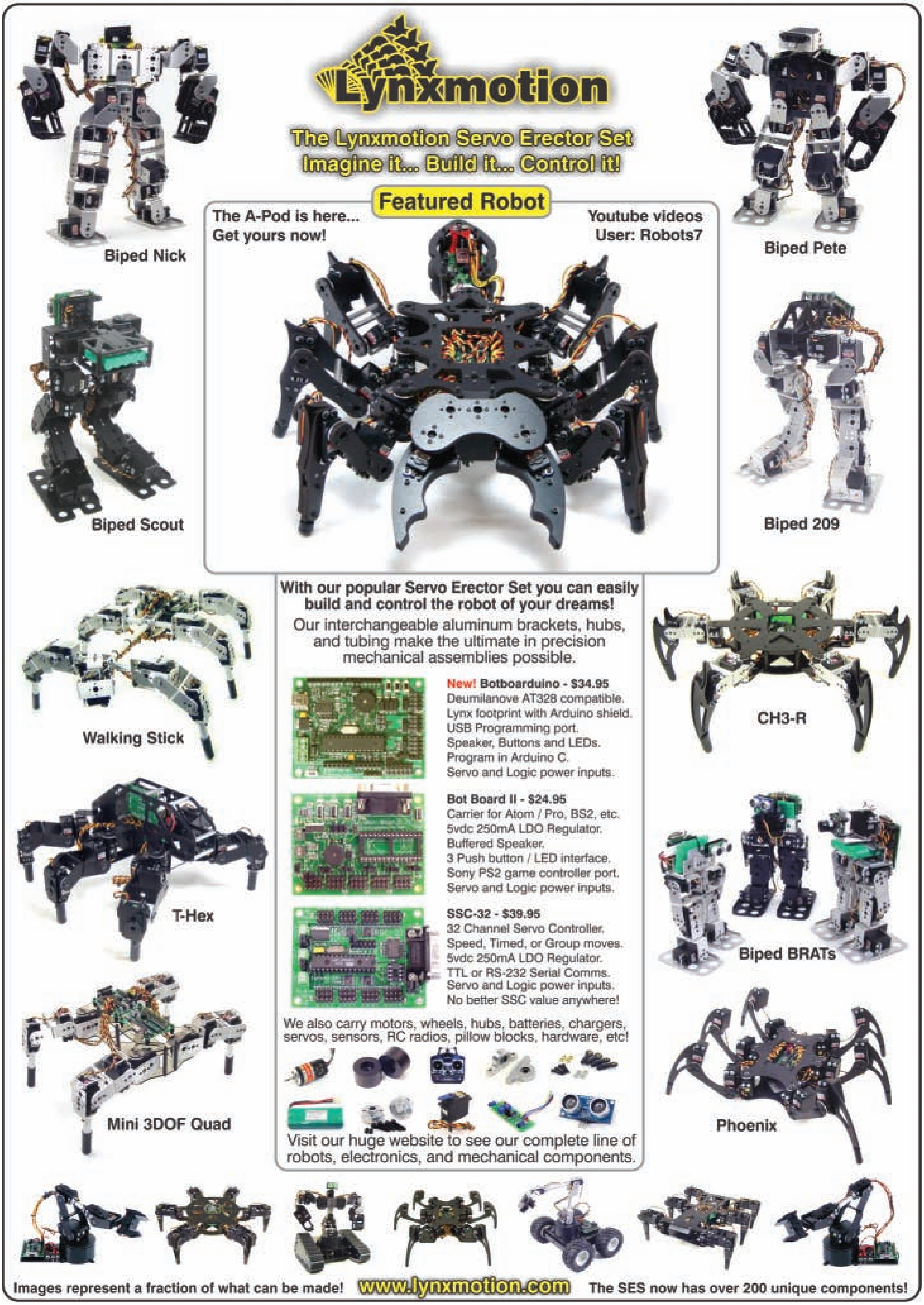
Other than avoiding apparently radioactive bananas, what robotic challenges does this inspire you to work on? SV

Resources

Video: Taylor Valtrop demonstrates teleoperation of Nao using Wii controllers, Kinect software, and other technologies.
www.youtube.com/watch?v=TmTW61MLm68

Video: Taylor Valtrop using teleoperation to manipulate the Nao to brush his cat, Lotus.
www.slate.com/blogs/future_tense/2012/01/04/nao_robot_brushes_cat_via_microsoft_kinect_video_.html

Taylor Valtrop's website.
http://taylor.veltrop.com


Lynxmotion
The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!
Featured Robot
The A-Pod is here...
Get yours now!
Youtube videos
User: Robots7
Biped Nick
Biped Pete
Biped Scout
Biped 209
With our popular Servo Erector Set you can easily build and control the robot of your dreams!
Our interchangeable aluminum brackets, hubs, and tubing make the ultimate in precision mechanical assemblies possible.
Walking Stick
CH3-R
New! Botboarduino - $34.95
Deumilanove AT328 compatible.
Lynx footprint with Arduino shield.
USB Programming port.
Speaker, Buttons and LEDs.
Program in Arduino C.
Servo and Logic power inputs.
Bot Board II - $24.95
Carrier for Atom / Pro, BS2, etc.
5vdc 250mA LDO Regulator.
Buffered Speaker.
3 Push button / LED interface.
Sony PS2 game controller port.
Servo and Logic power inputs.
T-Hex
SSC-32 - $39.95
32 Channel Servo Controller.
Speed, Timed, or Group moves.
5vdc 250mA LDO Regulator.
TTL or RS-232 Serial Comms.
Servo and Logic power inputs.
No better SSC value anywhere!
Biped BRATs
We also carry motors, wheels, hubs, batteries, chargers, servos, sensors, RC radios, pillow blocks, hardware, etc!
Mini 3DOF Quad
Phoenix
Visit our huge website to see our complete line of robots, electronics, and mechanical components.
Images represent a fraction of what can be made!
www.lynxmotion.com
The SES now has over 200 unique components!

Our resident expert on all things robotic is merely an email away.
roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

by
Dennis Clark

ASK MR. ROBOTO

It's April already! There are plenty of robot competitions and Battle-Bot-ish stuff coming up. Be sure to check out the Events Calender in this issue. Anyway, on to something interesting ...

Q. I want to reply to a previous issue when you talked about the chipKIT. I have been reading up on these quite a bit, and I really want to explore the chipKIT MAX32. I was wondering if you could do a more in-depth write-up on this. I would love to see more of the capabilities of this guy. I know Fred Eady has done some articles on it, but his are more advanced topics; I still consider myself a beginner/novice in the electronics world. Learning about CAN is still well above my head. Now, I'm not saying I want to see a "Hello World" program, but I would like to see some of the more basic stuff such as utilizing the ADC with sensors, basic timer/interrupts, and the like. Also, I would like to see it interfaced with MPLAB utilizing C instead of the Arduino processing structure. If I want help with that, I can just go to the Arduino site. I'm more interested in using this as a development board rather than a small projects board. Thanks!

— Corey Hastings

A. This is an interesting topic, to say the least. I've been wanting to investigate these boards a bit more, as well. There is a LOT of capacity in the ChipKit MAX32 board, and it has the side benefit of being Arduino compatible with the new Arduino MPIDE system. I "assume" that you have not hitched your wagon to the "Cult of Arduino" since you want to have projects for this board that use the MPLAB IDE instead of the Arduino. You will definitely have more flexibility and smaller (perhaps faster) code if you use MPLAB over Arduino, but even though you aren't a fan you should look into the value of the Arduino IDE (Integrated Development Environment).

The hardware objects written for the Arduino are abstracted enough to remove the drudgery of writing setup code and dealing with the arcane nuance that drives so many beginners mad when writing embedded programs. Try it and see if it fits your needs, then you can move on; some would say, step up to full-on embedded programming right at the "iron."

Because my ADHD tendency is to get sidetracked by new toys ("shiny things are nice!"), I'm going to start out here with some interesting servo code done with Arduino (I'll admit) just to shake out the hardware. Next month, I'll move on to an MPLAB example doing more stuff that folks can find interesting. The MAX32 board has easily accessible I²C and SPI interface pins, as well as a TON of other I/O that can be utilized. In fact, the MAX32 is only one board in Digilent's chipKIT continuum of PIC32 development boards.

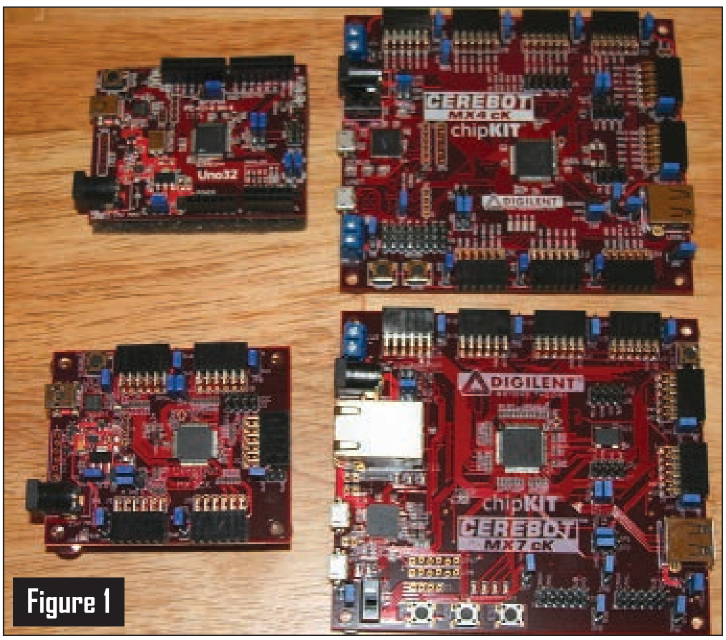


Figure 1

Figure 1 shows the full line of Digilent's chipKIT Arduino compatible boards. This includes the UNO32 —

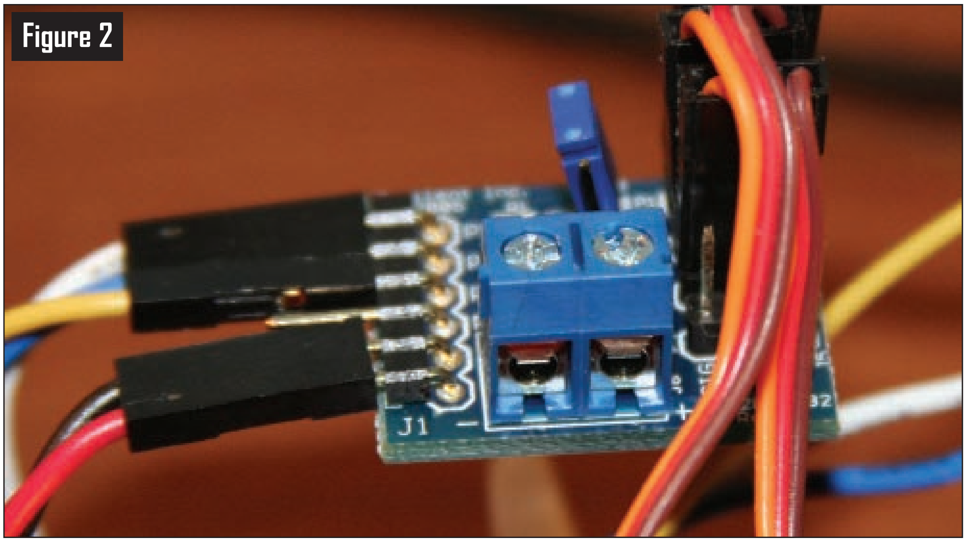
an Arduino UNO equivalent board that has 128K Flash and 16K RAM, and the CEREBOT MX3cK with the same specs and Arduino capability, but with the chipKIT pinout instead of Arduino. Also in **Figure 1** are the CEREBOT MX4cK (512K Flash/16K RAM) and MX7cK (512K Flash/128K RAM) boards. All are capable of being directly programmed by the Arduino or MPLAB. These boards range in price from under $30 to about $120. Their onboard development capabilities go up with the size and price of the board.

Okay. The boards are capable, look cool in the color red, and have lots of connectors, but how do we use them? The answer involves a bunch of wires. The CEREBOT boards have Pmod cables that nicely interface to Pmod boards (which Digilent has SCORES of!). However, the peculiar connector layout of an Arduino-based footprint does not lend itself well to using the Pmod boards directly. They can be used, but you'll have to use solderless jumper wires that have a six-pin female connector on one end and a bunch of individual female connectors on the other. Then, route the wires to power, ground, and signal lines as you require.

Like I previously mentioned, I am easily distracted and in the case of your question, got off on a tangent that involved making a six servo hexapod using nine gram servos that I got from *Hobby People* for $2 each (new!) and just HAD to build! *Hobby People* continually has outrageous sales on stuff like this; check them out at **www.hobbypeople.net** to see what I mean. They are an RC car and airplane distributor, but we hobby robotics folks use a lot of the same electronics and equipment in our playtime.

I am going to use six servos connected to the MAX32 via a PmodCON3 servo Pmod board — actually two of them — to drive my hexapod using an interesting idea that I got from my friend, Dan Michaels and his NICO robot

Figure 2



Listing 1: Some basic Arduino servo code.

```
// Variation of the Sweep example
// by BARRAGAN <http://barraganstudio.com>
// which is in the public domain.


#include <Servo.h>

Servo leftFront;  // create servo object to control a servo
Servo leftMiddle;
Servo leftRear;
Servo rightFront;
Servo rightMiddle;
Servo rightRear;

int pos = 0;    // variable to store the servo position

void setup()
{
  leftFront.attach(8);
  leftMiddle.attach(9);
  leftRear.attach(10);
  rightFront.attach(5);
  rightMiddle.attach(6);
  rightRear.attach(7);
}


void loop()
{
  for(pos = 0; pos < 180; pos += 1)  // goes from 0 to 180 degrees
  {                                  // in steps of 1 degree
    leftFront.write(pos);
    leftMiddle.write(pos);
    leftRear.write(pos);
    rightFront.write(pos);
    rightMiddle.write(pos);
    rightRear.write(pos);
    delay(15);
  }
  for(pos = 180; pos>=1; pos-=1)     // goes from 180 to 0 degrees
  {
    leftFront.write(pos);
    leftMiddle.write(pos);
    leftRear.write(pos);
    rightFront.write(pos);
    rightMiddle.write(pos);
    rightRear.write(pos);
    delay(15);
  }
}
```

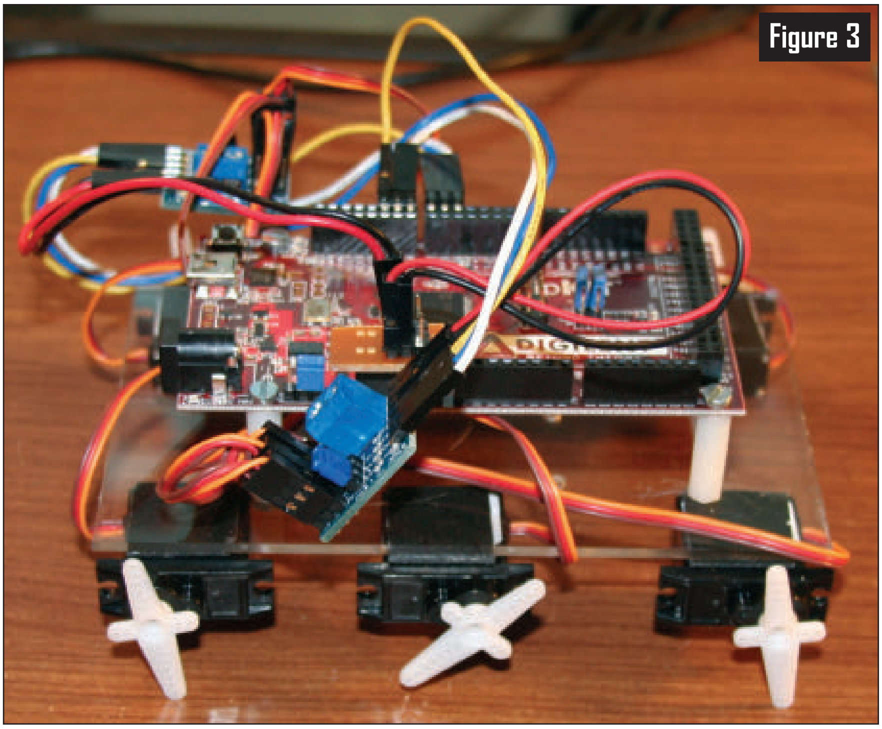


Figure 3

(**www.oricomtech.com/projects/legs.htm**). If you are into legged robots, you have to see Dan's site. You'd be hard pressed to find as much good information regarding legged robot gaits on a university site as you will on Dan's website! Anyway, you can see this Pmod board in **Figure 2** which is a close-up shot of one of the two boards that I used.

As I said, I'm using Arduino code here to illustrate integration of hardware and software to create a robot. So, in **Listing 1**, I have the basic code setup to run six servos from the board. I am using a Hobby People 7.4V, 850 mAh lithium polymer battery plugged directly into the MAX32 power jack through a homemade adapter cable. This will power a 1A 5V regulator on the board that we'll use to power the servos. The PIC32 is powered by a 3.3V regulator which appears to use the 5V regulator's output as its input. The board documentation warns that you should keep the voltage under 9V to avoid overheating the regulators if you decide to not use the 5V regulator (so you can put more voltage to the servos). I just used 5V; it is good enough to get everything working fine.

I'll admit, my robot in **Figure 3** doesn't walk with this example code, but it does a cool little dance! I built it by using RC hobby servo tape (which is *SUPER* sticky double-sided adhesive tape) to stick the servos on to a 7 cm by 12 cm piece of 2 mm thick clear Lexan. I'm using nylon stand-offs bolted to the board, and the Lexan to hold the rest together. The battery is under the board.

The whole project weighs in at about 200 grams which is less than half a pound. The wiring that I used came from other project boards and various connectors from the junk box – pointing out the value of a well-stocked junk box!

I did this project to create an experimental platform to learn on and have fun with by the time we finish our process here. Yes, even I will be learning something new here since I have not programmed this part before.

The big question is will I use my trusty ICD3 (you can also use a PICKit3 programmer) to program the board or stick with the Arduino bootloader. It works with the Intel hex file format that is the result of the Microchip C32 compiler, just like MPLAB would use. We can try both. It would be weird, but workable to use; for instance, try AVRdude in Arduino mode to download code via the USB serial port to the MAX32.

Next month, we're going to add a sonar and a Sharp IR rangefinder to our robot to give it eyes. I'll see what kind of Pmod boards or perhaps Arduino shields we can use, as well. Until then, stay tuned and check out the Digilent site (**www.digilentinc.com**) for more cool chipKIT products and add-ons.

As always, you can contact me at roboto@servomagazine.com to ask me any question about robotics that you have. I will do my level best to answer it! Until then, keep building robots and sending me questions. You're never too old to learn new tricks! **SV**

EVENTS Calendar

ROBOTS.NET

Send updates, new listings, corrections, complaints, and suggestions to: steve@ncc.com or FAX 972-404-0269

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**.

— R. Steven Rainwater

APRIL

12-14 **National Robotics Challenge**
Marion, OH
This year's event is the Canine Companion Challenge.
www.nationalroboticschallenge.org

14 **Brown IEEE Robotics Competition**
Brown University, Providence, RI
Here, 25 cm autonomous robots must navigate a maze.
http://brown.edu/Departments/Engineering/Organizations/Ieee/competition

14 **RoboRodentia**
California Polytechnic, San Luis Obispo, CA
Autonomous Micromouse-like robots must navigate a maze.
https://sites.google.com/site/calpolycomputerengineering

19-21 **VEX Robotics World Championship**
Anaheim, CA
Best high school and university VEX teams in the world compete.
www.vexrobotics.com/competition

20 **Carnegie Mellon Mobot Races**
CMU, Pittsburgh, PA
Events include Mobot Slalom and the MoboJoust.
www.cs.cmu.edu/~mobot

20-22 **RoboGames**
San Mateo Event Center, San Mateo, CA
Events include FIRA, BEAM, Mindstorms, and machine combat.
www.robogames.net

21 **Istrobot**
STU, Bratislave, Slovakia, EU
Line-following, IEEE Micromouse, Mini Sumo, and free-style events.
www.robotiks.sk

21 **Penn State Abington Fire-Fighting Robot Contest**
Penn State Abington, Abington, PA
Autonomous robots must navigate a maze and extinguish a fire.
www.ecsel.psu.edu/~avanzato/robots/contests

21 **Robot-SM**
University of Gothenburg, Gothenburg, Sweden
Events include Sumo, Mini Sumo, and Robot Pentathlon.
www.robotsm.se

21 **The Tech Museum of Innovation's Annual Tech Challenge**
Parkside Hall, San Jose, CA
Shake, Rattle, and Rescue — robotic rescue devices for earthquakes.
http://techchallenge.thetech.org

21 **UC Davis Picnic Day Micromouse Competition**
University of California, Davis Campus, CA
Micromouse maze solving.
www.ece.ucdavis.edu/umouse

23-24 **IEEE TERPA Student Robotics Competition**
Boston, MA
This year's theme is robotic cars.
www.ieeerobot-tepra.org/studentcomp.html

25-28 **FIRST Robotics Competition**
Edward Jones Dome, St. Louis, MO

FIRST teams from all over gather to compete at the annual championship.
www.usfirst.org

28 **Robotic Day**
Prague, Czech Republic
Lots of events including Bear Rescue, line following, Mini Sumo, and RoboCart.
www.roboticday.org

MAY

5 **RoboRave**
Albuquerque, NM
Events include fire fighting and line following for autonomous robots.
www.roborave.org

12 **Baltic Robot Sumo**
Riga, Latvia
Autonomous bots compete in Mini Sumo for a traveling cup award.
www.balticrobotsumo.org

12 **Western Canadian Robot Games**
Calgary, Alberta, Canada
Lots of events including Sumo, Mini Sumo, LEGO Sumo, Art Bots, line following, high speed line following, Minesweeper, LEGO Mindstorms Landslide!, as well as Humanoid and walker challenges.
www.robotgames.com

14-18 **ICRA Robot Challenge**
St. Paul, MN
A challenging event for robots that includes the Micro-robot Challenge, the Virtual Manufacturing Challenge, the Solutions in Perception Challenge, and the Modular and Reconfigurable Robot Challenge.
www.icra2012.org

2012 DIGILENT DESIGN CONTEST

The 8th annual US regional Digilent Design Contest will be held on May 6-7, 2012 in conjunction with the IEEE EIT conference in Indianapolis, IN. For further information, go to **www.digilentinc.com/events/ddc2012/** or **www.facebook.com/Digilent?sk=app_244940582228984**.

Robotics Showcase

NEW PRODUCTS

Super-Duty Pan System

ServoCity introduces their new P785SD super-duty, closed-loop pan system. By incorporating the HS-785HB servo, this super-duty pan system offers full positioning feedback. A 4:1 gear reduction enables the pan to rotate 380°, given a standard pulse-width range (1,050-1,950 usec). With up to 730 oz-in of torque and an aluminum framework, the P785SD is able to handle up to 30 lbs (balanced load). Dual ABEC-5 ball bearings support the 1" OD hollow stainless steel shaft which allows wires and cables to be routed through the axis of rotation. This system will work with any standard servo controller or radio system. It ships fully assembled with an HS-785HB for $159.99.

Pre-Cut ABS Sheets and Attachment Blocks

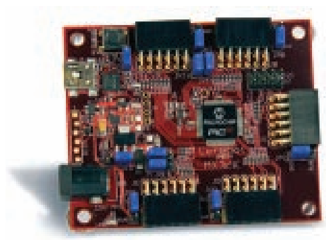
Also available from ServoCity are ABS plastic sheets which are perfect for custom projects. ABS (Acrylonitrile Butadiene Styrene) can easily be sheared, sawed, drilled, or formed to fit an application — whether you're building a skid plate for an R/C vehicle, a switch panel on a robot, or a trim plate for a car stereo. This material has excellent high impact resistance, high heat resistance, good dimensional stability, and has superior electrical insulation properties. ABS sheets have one textured (haircell) side and one smooth side. ServoCity is now offering pre-cut sheets in both 1/8" and 1/4" thicknesses; the price ranges from $1.49 to $6.99, depending on size.

ServoCity's new Attachment Blocks make it easy to join two flat panels — such as the ABS sheets — at a 90° angle. Constructed of 6061-T6 aluminum, these threaded blocks are extremely rigid and strong. Each block has two 6-32 tapped thru holes. These Attachment Blocks are exclusive to ServoCity and come in packs of 12 for $9.99.

For further information, please contact:

ServoCity Website: **www.servocity.com**

PIC32-based Cerebot Development Boards



Microchip Technology, Inc., and Digilent®, Inc., have announced several new 32-bit PIC32 microcontroller (MCU)-based Cerebot™ development boards with prototyping capabilities for the Arduino™ compatible chipKIT™ development platform. The Cerebot™ MX3cK, Cerebot MX4cK, and Cerebot MX7cK (MX3/4/7) boards provide a single, general-purpose development platform for users to create a wide range of 32-bit MCU applications using the free, Arduino-compatible chipKIT IDE called the Multi-Platform IDE, or "MPIDE." The Cerebot MX3/4/7cK boards break free from the traditional Arduino form factor, providing flexible pin access and connectivity with Digilent's line of Pmod™ peripheral modules.

Introduced in May 2011, the PIC32 MCU-based chipKIT Uno32™ and Max32™ boards enable hobbyists and academics to easily and inexpensively add electronics to their projects, even if they don't have an engineering background. The new Cerebot "cK" development boards include hardware that enables connectivity to the MPIDE, so users can develop with chipKIT via a bootloader application. Microchip's PICkit™ 3 debugger/programmer can be used with the Cerebot MX3cK. The Cerebot MX4cK, and MX7cK boards feature an integrated programmer/debugger. These boards are each populated with multiple connectors for Digilent's numerous Pmod I/O interface boards which provide ready-made interface circuitry for LCD, wireless, motor-control, sensor, and many other applications, minimizing the need for users to create original circuitry. The MPIDE includes libraries, such as Brian Schmalz's SoftPWMServo library — which enables users to generate an analogWrite-style output, as well as an RC servo output on all pins simultaneously. The Cerebot MX3cK board is priced at $39 each; the MX4cK board costs $79 each; and the MX7cK board costs $99 each.

For further information, please contact:

microchipDIRECT Website: **www.microchip.com/get/U007**

Electric Potential Sensor for Gesture Measurements

Saelig Company, Inc., announces the availability of the EPIC sensor (Electric Potential Integrated Circuit) — a

new, innovative electric field sensor. This new sensor technology measures electric field changes without requiring physical or resistive contact. EPIC is an award-winning, patent-protected sensor that can rapidly measure electric potential sources such as electrophysiological signals or spatial electric fields. The EPIC sensor simplifies the way movement sensing, proximity non-touch switching, or even gesture recognition signals are taken in medical and sports instruments, toys, electric appliances, smart lighting, gaming, and security. The electrode surface of the detector is coated with a passivated thin dielectric for direct application to a test surface without the need for electrically conductive gel. By detecting changes in the electric field, the EPIC sensor can drive a relay to act as a simple non-touch electric switch. It can be employed in a proximity mode or to detect specific kinds of movement as a gesture recognition device. Since the EPIC sensor does not need line-of-sight, it can detect movement through solid walls and can also be used to replace — or as an adjunct to — passive infra-red (PIR) sensors in a variety of applications including security motion detectors.

The sensor can be integrated on a chip with other features such as data converters, digital signal processing, and wireless communications capability.

For further information, please contact:

Saelig	Website: **www.saelig.com**

ELEV-8 Quadcopter Kit — Flying Robot Platform

The ELEV-8 quadcopter from Parallax is a flying robotic platform that is lifted and propelled by four fixed rotors. There are no fixed wings; all of the lift is created from the rotors. Unlike standard helicopters, a quadcopter uses fixed-pitch blades whose rotor pitch does not vary as the blades rotate; control of vehicle motion is achieved by varying the relative speed of each rotor to change the thrust and torque produced by each.

The quadcopter uses a HoverFly board with a Propeller multicore microprocessor to electronically control stabilization of the aircraft. The benefits to this system are a stable platform with no mechanical linkages for a small maneuverable and agile aircraft. The kit provides an inexpensive way to get involved in the quadcopter arena. The kit includes: frame, mounting hardware,

motors, speed controllers, propellers, and the control board for flight stabilization (the only thing you need to provide is the RC radio equipment and battery). The ELEV-8 platform is large enough for outdoor flight and has plenty of room for payload and attachments (up to 2 lbs).

Features include:

- Open under body for camera mounting options.
- Open source; design files available online.
- Highly modifiable for custom additions.
- Protected electronics.
- Common 4-40 hardware.
- Modular frame for fast repair.
- Protective motor mounts.

Price on the ELEV-8 is $550.

For further information, please contact:

Parallax	Website: **www.parallax.com**

bots IN BRIEF

THIS BOT IS HOT

This robot sure doesn't look like much, but if you give it a chance it'll be your new best friend on those cold and lonely nights as it stores up heat and re-emits it to keep you nice and toasty.

Under that featureless cubizoid exterior, Hagent — as the robot is so named — has some wheels, some sensors, and a big pile of phase-change material. Phase-change material (or PCM for short) is something that stores or releases energy when it changes from a solid to a liquid (or any other combination of phases) or vice versa. So, for example, let's say you've got a cup of coffee that's really, really hot. You could put some kind of PCM into it and the PCM would melt, absorbing the excess heat and making your coffee drinkable. Then, as the coffee cooled down, the PCM would re-solidify, releasing all that stored heat keeping your coffee warm for much longer. Sounds like a great idea, right?

Hagent takes that heat storage concept and mobilizes it for the purposes of using energy more efficiently and keeping you cozy when it's cold out. The robot can sense heat (like an oven, a fire, or anything else), and when it does it drives over and hangs out, letting its pile of PCM suck down as much energy as possible. Then, it'll follow you around, acting like a little pet space heater as its PCM re-solidifies, up until the PCM has emitted all of its stored-up heat. It's cute and it's mostly free, since all Hagent does is take heat that you've already produced and shift it around a bit.

Created by Andreas Meinhardt and Daniel Abendroth from Germany, Hagent is actually still at a prototype phase.

PACK IT IN

Boston Dynamics' BigDog just got bigger and badder. DARPA's already gotten on the horse and saddled up the bot with a bunch of luggage, and chased it out into the wilderness to see how it'll do.

DARPA says that "physical overburden" is one of the biggest problems facing soldiers today. We've got lots of great technology designed to keep warfighters safe and give them an advantage in combat, but all that stuff adds up to the point where having to lug around 45 kilograms (100 pounds) of gear is not unheard of. The job of the LS3 (Legged Squad Support System) is to act as a pack mule, carrying up to 181 kilograms (400 pounds) of gear so that the humans can take it easy for a change.

We already know that BigDog and AlphaDog are capable of negotiating steep and slippery terrain while heavily loaded, but DARPA's planning an 18 month battery of practical tests to make sure that the LS3 can get the job done.

Features to be tested and validated include the ability to carry 400 lbs on a 20 mile trek in 24 hours without being refueled, and refinement of LS3's vision sensors to track a specific individual or object, observe obstacles in its path, and to autonomously make course corrections as needed. Also planned is the addition of "hearing" technology, enabling squad members to speak commands to LS3 such as "stop," "sit," or "come here." The robot also serves as a mobile auxiliary power source — troops may recharge batteries for radios and handheld devices while on patrol.

The overall goal is for the LS3 to be able to behave functionally identically to a well-trained pack animal, albeit one that makes a lot of noise, eats gasoline, and can be used to recharge your iPod. If all goes well, the testing will culminate in a field exercise where the LS3 will embed itself with real live Marines. It'll be interesting to see how the soldiers will like a system such as the LS3, and whether the robot will be able to keep up with the demands of a realistic mission.

bots IN BRIEF

COFFEE, TEA, OR QB?

Do you need a receptionist at your company? Are you having trouble affording another employee? Do you like spending time with robots more than humans? If you answered yes to any (or all) of these questions, you might want to check out a brand new service being offered by Anybots that will solve all your problems.

Anybots' QB telepresence robot is famous for ordering scones, gambling in Vegas, taking repeated punches to the face, and hosting certain robotics blog editors. QB has a self-balancing mobile base with a bunch of sophisticated mobile videoconferencing hardware on top. Using a computer with an Internet connection, you can take control of a QB robot anywhere in the world, driving it around, seeing through its eyes, and interacting with people just like you were there yourself. Well, almost. That's the idea, anyway.

Now, Anybots is offering an entirely new service called AnyLobby that leverages the telepresence capabilities of QB to offer full-time telepresence staff to companies who might not otherwise be able to afford a real live human. Here's how it works: For about $2,400 a month, Anybots will send you a QB robot, then a professional human will log in to the robot and be available as a receptionist or an assistant for 40 hours a week. The human on the other end can be physically located anywhere with good Internet, and for locations with only intermittent need, one human can control multiple QBs, saving everybody time and money. **GetRobo.com** recently talked with one of these human receptionists (via a QB) named Angela:

"We can do a lot of things." QB doesn't have arms, but thanks to digital technology, she doesn't have any trouble scanning faxes and/or printing documents. The only thing she can't do is provide her signature when a package arrives, but the companies she works for have set up protocols for that — "Call Bob when there's a package."

Oftentimes, robots are thought of as something that can take away jobs, but Angela disagrees. "That is not the case here. It is creating jobs for small towns with high unemployment rates."

Anybots is hoping that these "virtual employees" that AnyLobby provides will offer flexible options for small companies who might not otherwise be able to afford a 100 percent old-fashioned home-grown human being. There's actually lots of potential here; it could be extended to other experts, as well. Need some on-site tech support? Just find a qualified person from anywhere in the world and they can have a physical presence right there with you immediately. Telepresence robots won't be completely replacing humans anytime soon, but if they can offer the advantage of being somewhere in person for a fraction of the cost and inconvenience, they might make a viable alternative in plenty of useful situations.

BIN THERE, DONE THAT

Jerome Mack ran a company that made robotic tools. In 1998, his best friend died in a grain bin accident and so he formed Mack Robots to make bots that could do this type of work. Last year, the $15,000-$17,000 Bin Bots began selling and were featured at a recent Minnesota Grain and Feed Association convention and trade show. The service bot measures 6 x 2 feet, weighs 800 lbs, and can fit through a grain bin door. It is also strong enough to lift heavy buckets. It can be operated by a worker from outside with an optional video camera and lights. Some highlighted features include:

- Remote control operation from outside the bin.
- Small enough to fit into any bin.
- Can push, pull, or lift a sweep and knock down a wall of grain.

- Dust-proof, heavy-duty metal body.
- Battery operated with 12 hours of charge life.
- Optional camera and lighting attachments.

GROUP EFFORT

Ecoland International, Inc., has formed the "Robotics Research Group" (RRG) which aims to develop state-of-the-art robotic solutions and partner with industry leaders to integrate those technologies, and to also provide innovative products and services. RRG's team of engineers and technicians will tackle some of the most difficult robotics and automation puzzles.

The modern robotics market has existed for nearly 30 years, but within the last decade substantial improvements in overall functionality, levels of control, and cost structures have been achieved. While many of the advancements in robotics have been in industrial markets where higher amounts of spending have allowed the development and commercialization of highly technical — yet costly — robots, many of the lessons learned are quickly trickling down to other market segments, including health care, business and commercial markets, and personal robotic devices.

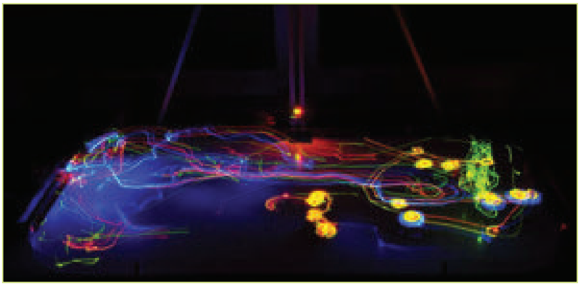
Market researchers are forecasting a US$3.6 billion jump in the industrial robotics market by 2015, from US$6.4 billion last year. The auto industry has traditionally been the largest single user of industrial robotics, but increasing use of robots in sectors such as food handling and processing, clean technology, and energy, as well as pharmaceutical and general consumer goods production will lead to increased demand for industrial robots as manufacturers look to improve the speed, quality, and reliability of production through automation. According to Japanese Robot Association research, the personal robot industry will be worth more than US$50 billion by 2025.

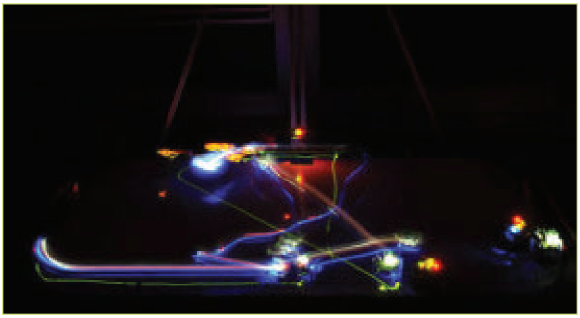
ICE QUEEN

When a Canadian gets their hands on a robot, you can be sure that one of two things will happen: Either they'll send it into space or they'll teach it to play hockey.

Since Darwin-OP has yet to be officially certified against either the harsh environment of outer space or guaranteed not to go crazy and kill a bunch of astronauts, it looks like this particular robot (who lives up at the Autonomous Agents Laboratory of the University of Manitoba) will just have to learn how to play hockey instead. Her name is Jennifer, and she might actually be the first autonomous humanoid robot ice hockey player in the world.

Jennifer's just a beginner and she's got a ways to go before she'll be able to convince anyone that hockey is a real sport. Getting a robot to skate isn't easy but it's certainly possible, and a pair of customized aluminum roboskates (currently on order) should help. The other tricky bit is the aiming and shooting. Darwin already comes with ball tracking and the ability to aim kicks at a goal, but using a hockey stick to aim a puck at a goal sideways is an entirely different skill. Aside from being what looks like a lot of fun, this project is a submission to the Darwin-OP Humanoid Application Challenge which is scheduled to take place at the 2012 IEEE International Conference on Robotics and Automation (ICRA) in Minnesota this May.





ROBOCKEY JOCKEYS

The final exam in University of Pennsylvania's Design of Mechatronic Systems class has the students designing a mechatronic system in the form of a small team of autonomous hockey-playing robots. These teams of robots face off against each other in a big tournament of "robockey" (not robohockey, but robockey) where we'll assume the winning team gets an A+ and everyone else fails the course.

Sometimes these robockey matches take place in the dark, with admittedly cool results like the photo here.

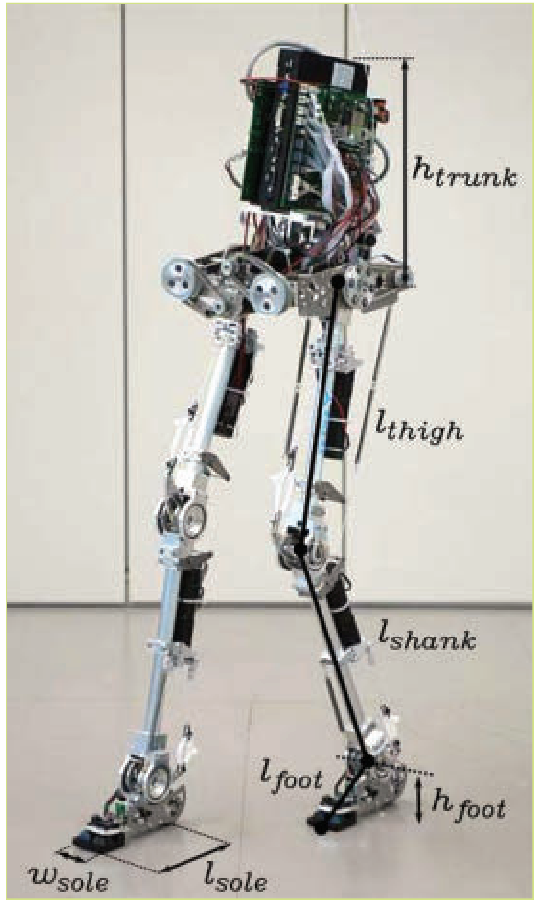
HAPPY WITH HOPPING

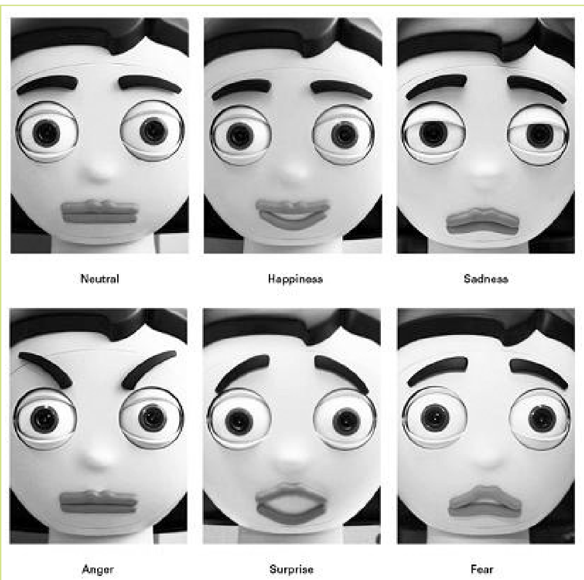
Researchers at the SIM Group of TU Darmstadt and the Locomotion Laboratory of Jena University are working on an ambitious new project they hope will display heel-to-toe walking like Boston Dynamics' PETMAN.

They're planning to build a series of evolving BioBipeds — musculoskeletal robots that incorporate elasticity to achieve energy-efficient standing, walking, and running.

They've only built the BioBiped1 so far, but it is already displaying some impressive hopping capabilities. This kind of hopping produces the same sort of stresses as a light jog, so it is a pretty good indication that the robot will be jogging on a treadmill in the future.

Each leg has four joints (hip x2, knee x1, ankle x1) actuated by a combination of both active motors and passive springs which are based on the muscles and tendons found in a human leg. The springiness of the legs means the robot can passively rebound when it lands on its feet. Eventually, a more sophisticated upper body with arms may be added, and various foot mechanisms will be evaluated. Currently, the robot is restricted to moving up and down, but future versions will gradually introduce more freedom until it is able to stand on its own.



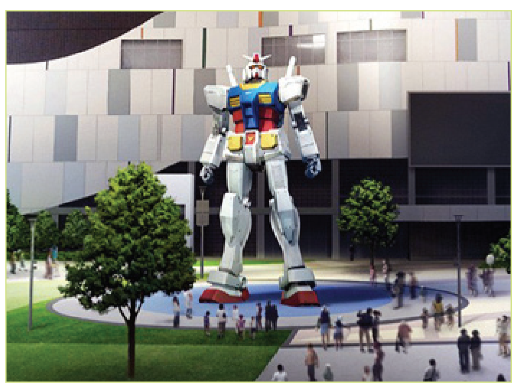


FACE IT

CITEC Bielefeld's anthropomorphic robot head "FloBi" has been upgraded with a cheap and simple motion-capture setup. You'll recall that FloBi is an expressive head with modular parts that allow you to insert different sections of the head to create a male or female robot.

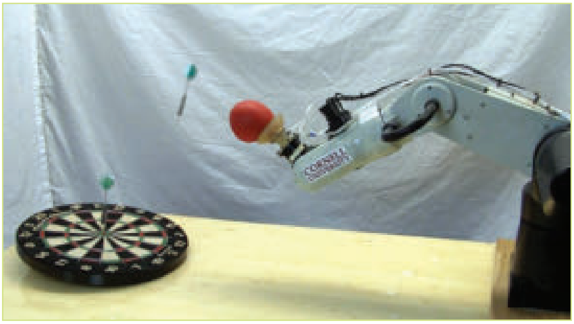
The mo-cap setup is a helmet with a single camera pointed at your face. It tracks your eyes, eyelids, brows, and mouth using color markers while an X-IMU (inertial measurement unit) detects overall head rotation. The recorded motion can be viewed using a virtual model or it can be played back on the actual robot. This means rather than having to animate each segment of the robot's face by hand (which can also lead to unnatural expressions), all of the motion can be recorded quickly and easily from a real person.

The result is a robot face that — despite being simple in appearance — is convincingly lifelike due to its realistic eye movements. The lips aren't quite malleable to accurately recreate lip-synching, but given the technical limitations of the robot head and the simple motion-capture solution they've created, I'd say they have been pretty successful.



HOMEWARD BOUND

Japan's Gundam Mini-Theme Park is finally opening its doors April 19 at Diver City Tokyo. The giant Mobile Suit RX-78 will find a permanent home there, as well as a dome theater for showing videos, photo ops, and displays. The entire park measures about 2,000 sq meters and includes some Bandai gift shops and café.



WE BE JAMMIN'

The jamming robot gripper has learned a new trick! Roboticists at Cornell and the University of Chicago have taught it to throw stuff.

A quick refresher: The gripper is simply a latex balloon filled with coffee grounds. The grounds move around each other like grains of sand and can conform to objects and complex surfaces, but when air is pumped out of the balloon, the grounds all "jam" together into a solid mass, yielding a strong hold on whatever the gripper is in contact with. It's simple, cheap, and you can pick up just about anything without having to calculate optimal grasping points (or anything in the way of sensing or computation). You really just stuff the gripper against an object, pump the air out, and off you go.

This new "shooting" trick (or "fast ejection," if you prefer) comes from rapidly re-inflating the gripper with air. It sounds simple enough, but what you don't expect is the repeatable long-range accuracy — good enough to shoot baskets, sort hardware, and play a mean game of darts. Researchers say the precision they can achieve is ±60 mm with 95 percent confidence in the direction perpendicular to flight, which "is certainly too coarse for high-precision manufacturing tasks, but could be useful for tasks like sorting objects into bins in a factory or throwing away trash in a home."

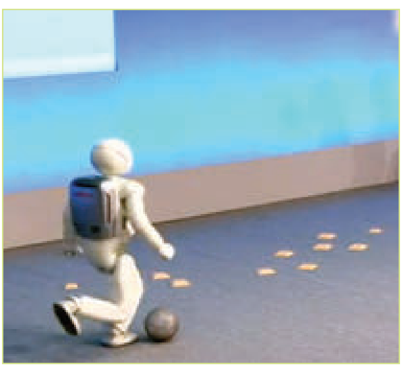
It's obviously good enough for winning games of mini-basketball and playing horizontal darts, so it's kinda fun to imagine what other tasks a talented throwing robot might do around the house ... like making a sandwich or unloading the dishwasher.

EYES ON OPTI-TRACK

In autonomous robotics, the right inputs reveal the right path. Each day, the University of Waterloo's WAVELab Research Group navigates these waters as they work out the best ways to help robot commanders get their flying mecha-assistants from point A to point B without the use of manual controls. For them, the most precise way to achieve this is with the help of an OptiTrack™ motion capture system.

Professor Steven Waslander began WAVELab's autonomous vehicle program with the traditional measurement tools of the field: IMU sensors, gyroscopes, and onboard GPS. These devices gave real time estimates that stabilized his squadron of aerial and ground rovers, but failed to deliver the control needed to prevent those same vehicles from veering off into hazardous areas (like walls). To address this problem, Waslander increased his state estimation system's scope by enlisting the input of an 18 camera OptiTrack tracking system.

The addition of the cameras enabled the team to estimate in real time if the onboard sensors were working and if the robots had stayed locked to their predetermined route. Quick adjustments to the onboard control algorithms could be made from this data which significantly reduced its development time. Now, each time the WAVELab's robots go for a spin on the ground or in the air, they can sense unknown surroundings, develop accurate 3D maps, and navigate through their environment with the assistance of OptiTrack's sub-millimeter data. As OptiTrack's system helps the algorithm increase in potency, new opportunities for robot crossover could start to materialize in our lives, as well. In time, you may see WAVELab's robots out in the world as sample collectors for scientists exploring hard-to-reach crevasses in Antarctica, or as scouts that keep soldiers safe. Maybe someday they could even be the pilot that flies you home from a well deserved vacation.



SOCCER IT TO US

This is the official goal of the RoboCup soccer competition: "*By mid-21st century, a team of fully autonomous humanoid robot soccer players shall win the soccer game, complying with the official rule[s] of the FIFA [Fédération Internationale de Football Association], against the winner of the most recent World Cup.*" There's been a lot of improvement over the last few years, but nothing that compares to the skills that the new version of ASIMO recently displayed (and RoboCup itself isn't far behind). ASIMO, of course, costs a ton of money and has the corporate support of Honda. However, watching RoboCup competitions themselves, you can see improvement that's almost as dramatic, albeit with a delay commensurate with the amount of time and money that can be invested in what's ultimately a hobby/research for most of the teams involved.

GLOVE GRABS AWARD

The SEM (Soft Extra Muscle) Glove from the Swedish company Bioservo Technologies is the winner of the Robotdalen Innovation Award 2012 and will receive support from Robotdalen for further product development and commercialization. A lot of people need extra strength to grab things in their everyday life, like a coffee pot or a drill. This is due to things like age, repetitive strain injuries, or medical conditions such as a stroke or arthritis. The SEM Glove adds extra power to the grip through force sensitive sensors and robotics technology.

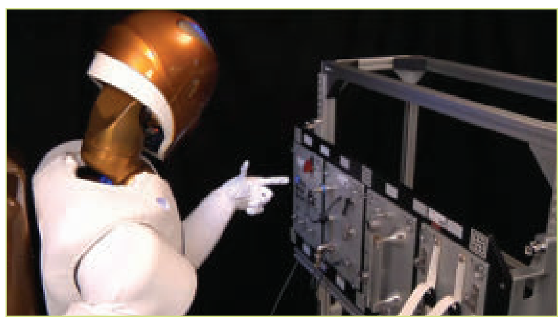
Winners of the Robotdalen Innovation Award 2012 (from the left): Mohammad Reza Ghahremani, Iran (2nd prize); Johan Ingvast from Bioservo Technologies (1st prize); and qbrobotics from Italy (3rd prize). Photo by Terése Andersson.

Bioservo Technologies was founded by researchers from the Karolinska Institute and the Royal Institute of Technology in Stockholm, Sweden. They combine their experiences about human needs with modern robot technology knowledge to create innovative products that strengthen the body. The SEM Glove is viewed as an excellent example of a user-friendly, adaptive robotic assistance. The prototype appears to be ready for serial production with great opportunities to reach a global market.

The Robotdalen Innovation Award aims at giving the winners an opportunity to further develop their commercially valid robotics solutions, says Lennart Karlsson, Internationalization Manager at Robotdalen. The winners were selected among 24 contributions from 12 countries.

Johan Ingvast from Bioservo Technologies demonstrates the SEM Glove. Photo by Terése Andersson.

PRACTICE WILL MAKE PERFECT



Robonaut 2 has been up on the International Space Station for a while now, and it's only recently that he's really gotten to wake up, stretch out, and get to work. What work is that? Well, it's not hand-to-hand combat with invading aliens. (Not yet, anyway.)

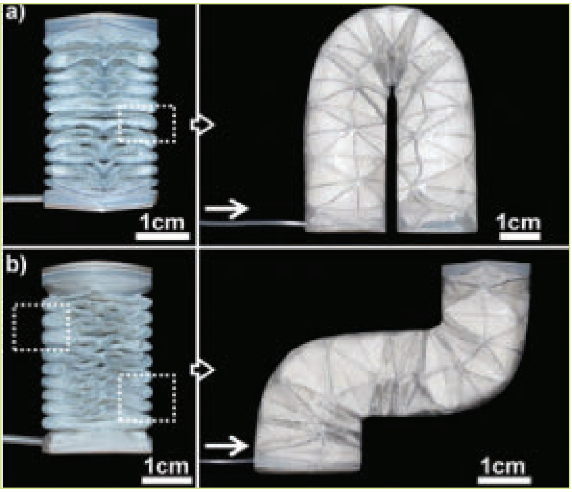
It's predominantly a range of motion tests to make sure that all of Robonaut's motors and joints are working properly, and aside from some fabric binding around the elbows, the robot checked out a-okay.

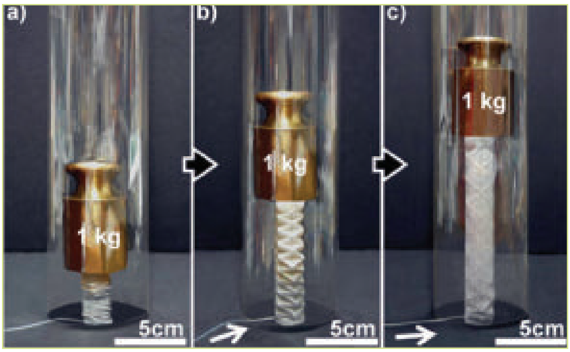
The other purpose of the motion tests was to try and figure out how Robotnaut's movements change in zero g as opposed to one g (one g is equal to the force of gravity on the Earth's surface). His limbs all still have mass, of course, but since they don't weigh anything, calibrations are going to be necessary to make sure that the bot retains all of that manual dexterity he's so well known for.

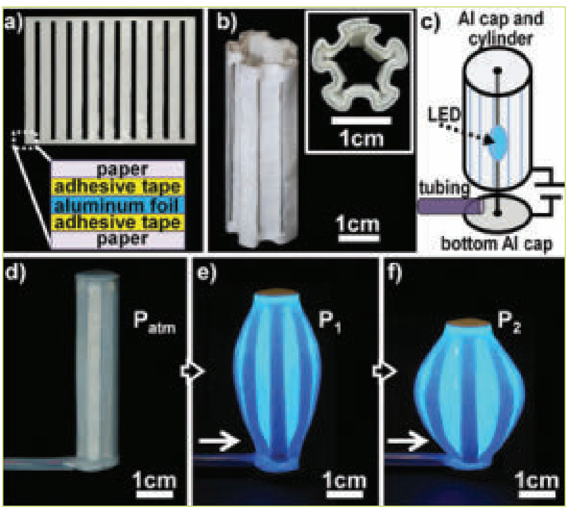
So, what's next (besides the fact that Robonaut can move his limbs)? Practice is what Robonaut is going to start doing next, using a taskboard where he can press buttons, flip switches, and use tools without risking accidental thruster firings, unexpected decompression, or arming of the railgun turrets or laser cannons.

Taskboard manipulation is not the most exciting of jobs, but it's an important first step in being able to let the robot perform autonomous tasks safely and reliably.

Cool tidbits herein provided by Evan Ackerman at www.botjunkie.com, www.robotsnob.com, www.plasticpals.com, and other places.







BEND IT LIKE ... PAPER

Remember that freaky air-powered boneless robot? Well, the same group that unleashed that thing on the world (George M. Whitesides' lab at Harvard) has started to manufacture some beautiful (and superbly functional) air-powered Origami robotic actuators out of paper and elastic. These "soft pneumatic actuators" are constructed by combining paper with a silicone elastomer called Exoflex in a mold and casting the composite so that they contain internal pneumatic networks. The process is fast, easy, inexpensive, and repeatable. When the pneumatic networks are inflated with an external source of air, the elastomer expands, creating an actuator.

The key to the funky shapes that these actuators can make is to use paper to constrain the ways in which the elastomer can bend. The simplest form of this is to just place a piece of paper along one side of the elastomer such that when it's inflated, it bends the other way. However, there are lots of creative things that you can do with paper. The examples in the top photo show pneumatic actuators made of accordion-folded paper/elastomer composites that have had certain folds glued together to generate specific shapes when they inflate. The second photo shows an extension actuator lifting a one kilogram weight (2.2 pounds) which is over 100 times the weight of the actuator itself.

Without the weight on top, an actuator like this can fully deploy at approximately the speed of sound. The researchers have also tested structures that can contract, twist, and even act as little lanterns by controlling the amount of light emitted by an embedded LED through aluminum panels.

The last graphic illustrates how it's also possible to embed electronics in these actuators, suggesting the possibility of creating an entire robot from little more than paper, silicone, and wiring. You'd need a pressure source too, but it might be possible to use chemical reactions to generate gas from relatively small amounts of liquids that could themselves be stored in flexible containers within the robot.

The advantages of robots constructed with methods like these are numerous: They're simple to make, flexible, expandable, lightweight, and cheap enough that you could make a whole bunch of them if you wanted to. There's a reason that DARPA has a whole program dedicated to soft robots: They can do all kinds of things (and go all kinds of places) that rigid robots just can't, and with actuators like these, nothing will be able to stop them.

GRASP ON QUADROTORS

The GRASP (General Robotics, Automation, Sensing, and Perception) Lab at the University of Pennsylvania is already famous for its quadrotor tricks, including bots that can fly through windows and hula hoops, build structures, and even land on each other. Now, those big bad quadrotors have been shrunk down into much smaller "nano quadrotors," and the GRASP Lab has been playing around with lots of them. One possible advantage of having smaller quadrotors is that you can cram more of them into a given space, allowing you to perform more complex swarm behaviors. These little beauties are eminently tossable, and it sort of looks like you can just chuck them like ninja stars and they'll self-right and stabilize themselves. Up to 20 of these things can fly in formation all at once, and it's very impressive to see them make a series of different 3D shapes.


UNIQUE PRODUCTS - FREE TECH SUPPORT!
BEST SELECTION AND LOWEST PRICES!
60MHz Digital Scope
2-ch 60MHz 500MS/s sample rate + 8" color TFT-LCD. Includes a FREE carry case! $329
SDS6062
Spectrum Analyzer
Handheld 2.7GHz for meas. radar, radio/tv, WiFi, Bluetooth etc. transmissions. $1,887
PSA2701T
SPI Host Adapter
Serial protocol host adapter. 32MB buffer 100MHz signal 8-bit pattern gen. $1,299
SPI STORM
60MHz Scope + L.A.
2GS/s scope, 200MS/s logic analyzer, 100MS/s pattern generator plus TDR. USB-pwrd. $239
MSO-19
USB to I2C
This "drop-in" solution connects a PC to I²C/SMBUS + 32 I/O lines. 'Solid' software. $89
USB-I2CIO
100MHz Scope
2-ch 100MHz 2GS/s scope w/ 2000 wfm/s refresh rate. Plus free carry case! $399
DS1102E
iPhone/iPad Scope
1-ch, 5MHz mixed signal scope adapter for iPhone, iPad and iPod Touch. $297.99
iMSO-104
XP Emb. Touchpanel
Touch-input 10.2" TFT color LCD 12V-powered Windows PC. LX800/512MB/4GB/SD. $699
CUPC-P80
25MHz Scope
2-ch 25MHz full color FFT-LCD scope with autoscale and trigger delay functions. $279
PDS5022S
World's Smallest
2-ch, 200kHz mixed signal scope on DIP module plus spectrum analyzer & AWG. $49
XPROTOLAB
LOW PRICES, FREE TECHNICAL SUPPORT & GREAT CUSTOMER SERVICE
Saelig .com unique electronics
WHY BUY ANYWHERE ELSE?
WWW.SAELIG.COM
FOLLOW US:


From Smiley's Workshop
An Arduino Workshop
Book: An Arduino Workshop $44.95
Projects Kit $84.95
Book and Kit Combo $124.95
C Programming for Microcontrollers
Book: C Programming for Microcontrollers $44.99
Projects Kit $58.49
Book and Kit Combo $99.99
Breadboardable LCD Navigator
Parts Kit for Smiley's Workshop
Breadboard friendly LCD and five-button keypad parts kit.
$34.95
Simple Chaser Lights Board
Parts Kit for Smiley's Workshop
LED Chaser Lights parts kit.
$24.95
www.nutsvolts.com
800 783-4624


Extreme Motor Speed Control!
MegaMoto Plus Shield
High-current motor control for Arduino™
♦ 7V-28V - H-bridge or Dual Half-bridge
♦ 30A Continuous with fan - 40A peak
♦ Current sensor outputs can be sent to any analog pin
♦ Stack up to three units on one Uno or compatible
♦ Independent half-bridges can control brushless/steppers too!
♦ Over temp and over current protection
Scorpion XXL
♦ 7V - 28V
♦ Dual 40A+ Peak H-Bridges
♦ Current/Temp limiting
♦ R/C inputs w/mixing
♦ Optional enclosure
♦ 3.25" x 2.25" x .5"
MADE IN THE USA
Check out our other motor controls
Dalf Motion Control System
♦ Closed-loop control of two motors
♦ Full PID position/velocity loop
♦ Trapezoidal path generator
♦ Windows GUI for all features
♦ Giant Servo Mode!
♦ C source for routines provided
♦ PIC18F6722 CPU
See www.embeddedelectronics.net
H-bridges: Use with Dalf or Stamp etc.
OSMC
♦ Monster power!
♦ 14-50V 160A!
♦ 3.15"x4.5"x1.5"
♦ 3 wire interface
Simple-H
♦ 6-28V 25A!
♦ 2.25"x2.5"x0.5"
♦ 3 wire interface
♦ current & temp protection
Magnum775
♦ RS-775 motor
♦ planetary gearbox
♦ 20:1 ratio - 700 rpm @ 14V
♦ Nearly 700W!
♦ Build something - rule BotsIQ!
OEMs ask about custom versions
ROBOT POWER™
www.robotpower.com
Phone: 253-843-2504 ♦ sales@robotpower.com


THE NEXT GENERATION OF MAXSONAR®
The HRLV-MaxSonar Sensors
• Amazing One-Millimeter Resolution
• Simultaneous Multiple Sensor Operation
• Superior Rejection of Outside Noise Sources
• Target Size Compensation for Accuracy
• Temperature Compensation ($4.95)
• Outputs now include TTL Serial
$34.95 (MSRP)
MaxBotix®
High Performance Ultrasonic Rangefinders
www.MaxBotix.com
facebook
THEY SAID IT COULDN'T BE DONE SO WE DID IT

COMBAT ZONE

Discuss this section in the *SERVO Magazine* forums at http://forum.servomagazine.com

Featured This Month:

www.servomagazine.com/index.php?/magazine/article/april2012_CombatZone

BUILD REPORT:

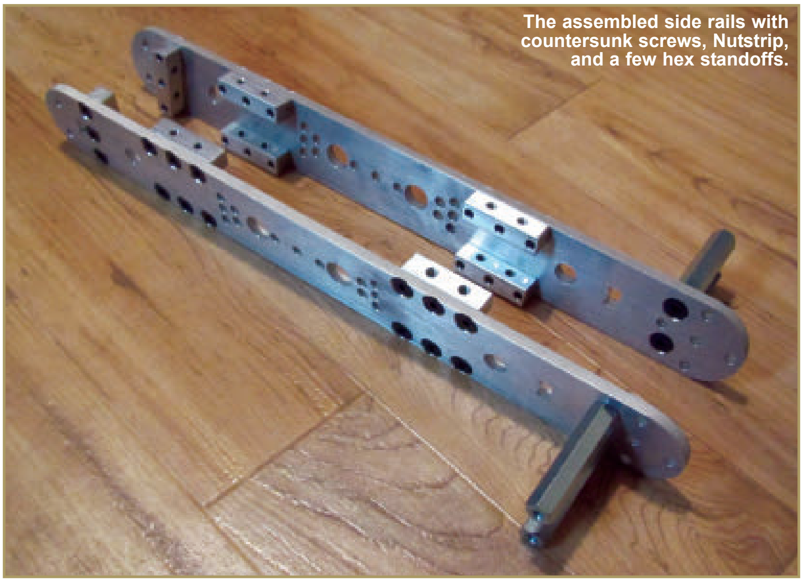
*Nyx, Sportsman, and Dragon*Con 30 lb Bot – Part 2*

● by Mike Jeffries

Monday

With a fresh pile of parts and Motorama just over a month away, I got a quick jump on assembling Nyx. The first part that was assembled was the weapon hub which also led to the discovery of the first problem. The 3D model of the sprocket for the weapon supplied by McMaster-Carr had the wrong hole pattern. With the spacing off, only three of the six holes would be usable.

This problem would have to be shelved until I picked up a 0.201"



The assembled side rails with countersunk screws, Nutstrip, and a few hex standoffs.

drill bit to drill a new set of clearance holes in the sprocket. The other task for Monday was countersinking all of the holes in the inner side rails and mounting the Nutstrip that I got from **Kitbots.com**.

Tuesday

There was not a lot of progress on Tuesday, because I was waiting for a parts order with the 0.201" drill bit I needed to modify the weapon sprocket, and proceeding with assembly would require a good portion of the work to be undone to get it drilled out. I did get two small but important parts done: I cut and glued a rubber shock pad to the area on the front of the chassis that would be hit by the attachments, and I modified the keyway on one of my DeWalt shafts to allow the sprocket to sit in the location it needed to.

I tossed a 1/8" end mill on my mill, clamped the shaft as best I could, and dropped the bit into the existing keyway groove. With a slow spindle speed and ample cutting fluid, the new keyway was easily added to the shaft. In addition to this, the bearings for the weapon were glued into the bearing blocks and round weapon hubs.

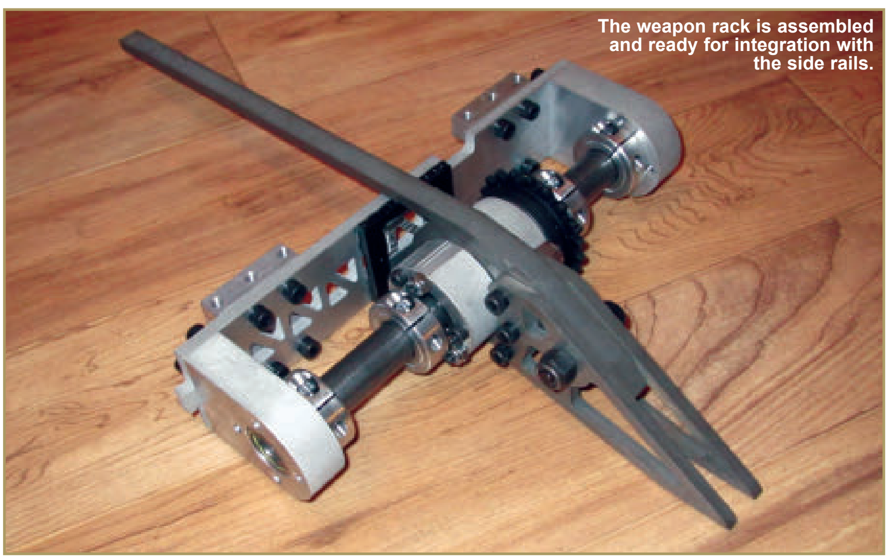
Wednesday

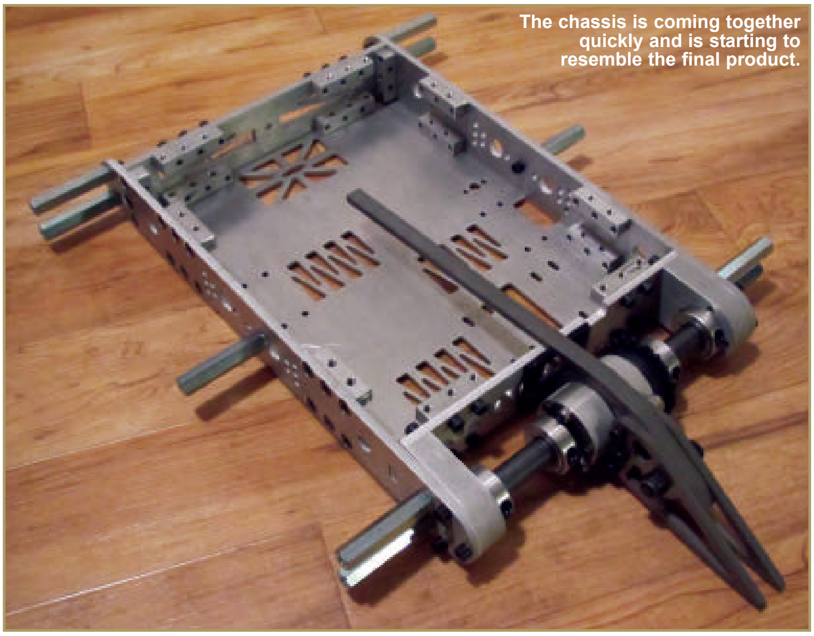
The first task of the evening was to drill out the sprocket and get the weapon rack assembled. Once again, a slow cut with ample cutting fluid resulted in a quick and easy modification to the steel part.

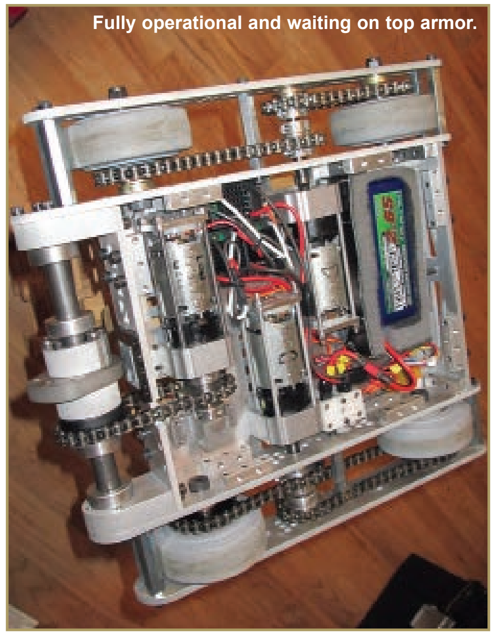
The next step in the build was assembling the weapon rack. Stage 1 was assembling the weapon hub which consists of the main 4130 steel universal attachment block; the two outer aluminum rounds are aligned with it; and three long 10-24 bolts are run through the length of the parts. I next added the sprocket and used the existing tapped holes to pull the assembly together. The other three bolts were added with locknuts on the end,

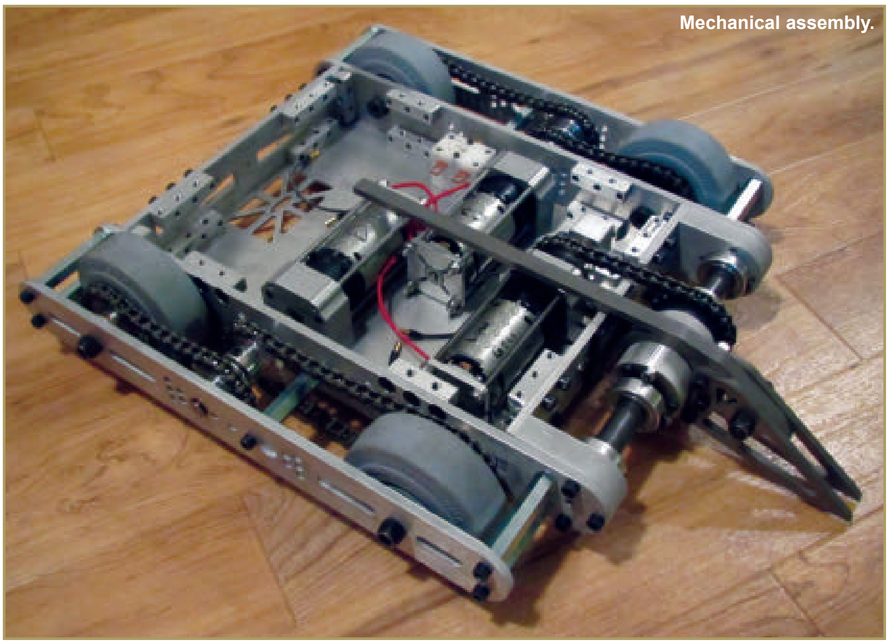

The modified shaft is next to a stock shaft showing the difference in the keyways.


The 19 tooth sprocket after modifications.


The weapon rack is assembled and ready for integration with the side rails.


The chassis is coming together quickly and is starting to resemble the final product.


Fully operational and waiting on top armor.


Mechanical assembly.

and the main axle was slid into the assembly. The outer bearing blocks were slid on and the front armor plate was bolted to the blocks.

With a bit of sanding and hammering, the side panels were coaxed onto the front and rear armor panels which were quickly fastened together. The remaining hex standoffs were added, followed by the baseplate which had mistakenly been modeled with 0.250" holes for all of the 1/4-20 bolts. The snug fit was worked around and the CAD drawings were corrected.

Thursday

This was the dirtiest day of the build. The wheels and outer rails were added first, with washers being used to position the wheel sprockets at safe distances from the frame rails. The next parts to go in were the three 18V DeWalt PowerDrive Kits from **www.robotmarketplace.com**, along with the X spline shafts and bushings in the outer rails.

A combination of sprockets, shaft collars, and washers were added to the shafts to position the 7075 aluminum sprockets in alignment with the wheel and weapon sprockets.

After alignment was completed, it was time to add roller chain and wrap up the mechanical assembly. I wrapped the individual sprocket segments in roller chain to determine where I needed to break the chain and used a commercial chain breaker to split the segments.

Once this was completed, each chain was wrapped around the corresponding sprocket set and master links were added to keep the ends together.

Saturday

After taking Friday off, it was time to tackle the electronics section. The wiring in Nyx is fairly simple. Each motor is controlled by a Holmes Hobby BR-XL speed controller; they all run to common leads — the positive to a Team Whyachi MS-01 switch I had left over from an old robot, and the negative to the negative lead of the battery plug. The power leads connect to a

Turnigy 2.65 Ah 6s lipoly pack that provides power for the robot.

After several hours of soldering, crimping, and cutting, the electrical system was done. A quick calibration later and Nyx was driving.

I chose not to have the top armor included in the first waterjet order because I wanted to be able to pick the most durable armor the remaining weight afforded. At this point, Nyx weighed enough that with the current armor plans it would end up overweight by around 0.5 lbs. The design for the top armor was modified, redundant screws were removed, and the steel threaded rod on the weapon gearbox was replaced with the same aluminum that the drive motors use.

The completed Nyx prior to Motorama with all three weapon options shown.

Closing Remarks

After a quick weigh-in, paint job, and reweigh just to be certain, Nyx was completed and weighed in at 29 lb 14.8 oz with the heaviest weapon option. The lifter weighs 20.2 oz, the spike weighs 14.9 oz, and the dual C shaped spinning disks weigh 9.6 oz. **SV**

The History of Robot Combat: The Rise of the Insects – Part 1

● by Morgan Berry

This article was difficult to put together, to say the least. Once again, the robot combat builders haven't proven to be the most diligent record keepers (with one notable exception). To our good fortune, though, the wonderful community racked their brains and pulled together a wealth of information on the development of the Insect class. So much information, in fact, that I've split the Insect discussion into two articles: one on Antweights and one on Beetleweights. As always, if you notice any holes or inaccuracies in our story, please contact SERVO *and let us know. With your help, we'll sort out the soap opera that is the history of robot combat.*

S.E.C.R. BATTLEBOTS BATTLEANTS

For most of robot combat's history, the mantra was "the bigger, the better." In both Robot Wars and BattleBots™, the smallest weight class was 60 lb and with good reason. Many of the most memorable bots and some of the most exciting matches were from the Superheavyweight (340 lb) class. However, as these massive television spectacles began to end, the priorities of the sport changed. As I discussed in last month's installment (*The History of Robot Combat: Life after BattleBots*), in the years immediately following the cancelling of BattleBots, the spirit of the sport changed to become even more grassroots and inclusive to new builders. Let's face it; for many, even a 60 lb robot can be both too intimidating and expensive to build

The author driving Babe The Blue Bot at the 2003 "The Capital Offense" event in Tallahassee, FL.

The author and the rest of Legendary Robotics around the Insect arena at TCO.

(recall Combat Zone editor Kevin Berry's massive flop at Robocide in 2003, also discussed last month). Enter the Insect weight classes. The most popular Insect classes are Antweight (up to one pound) and Beetleweight (up to three pounds). In addition to these are the 150 gram Fairyweight and the six pound Mantisweight. These scaled back bots were much less expensive and somewhat safer to build (although I will again remind you: All robot building is dangerous! And, while the blade on an Antweight might not take off your hand, it could easily take one of your fingers, so follow all safety precautions when building), making the Insect class attractive to new and inexperienced builders. As a result, in 2003 (the year after the end of BattleBots), the number of Insect competitions across the United States exploded. In this new era of robot combat, the Insect had become king.

Before I get into the history of the Antweight, I'd like to share my personal experience with the class. It was around this same time period that I was introduced to robot combat. My first event was at the SECR-sponsored The Capital Offense in July '03. I was nine years old, and shared an Antweight wedge called Babe the Blue Bot (a play off of Paul Bunyan's companion Babe the Blue Ox) with my seven year old brother. I lost in the semifinals to Cuzin It, driven by Ross Embry from TeamPyramid, due to mechanical failures by Babe.

In the true spirit of the sport (and maybe because he just felt bad about beating a team of small children), Ross offered a friendly rematch to me, and I walked away with the pride of knowing I could have won the semifinal match if not for the mechanical troubles.

I was pretty regularly involved in the SECR matches in Florida from that point on, and in that time I discovered that the true kindness extended to me by Ross was not unique; it was simply how the builders treated each other. I also learned that, even as a little girl, I could hold my own against much older competitors (although, looking back on it, they may have been reluctant to aggressively compete against a 10 year old girl), and won my fair share of matches. My experience with robot combat was one of the most fun, exciting, and enlightening that I have ever had, and I look back on it with great fondness.

Readers who have been following this series might remember the importance of Denver's Critter Crunch in the establishment of the sport. In addition to this, Critter Crunch can also claim a critical role in developing the Insect class.

Although Critter Crunch — the mother of all robot combat competitions at MileHiCon in Denver, originally in 1987 — had no specified weight classes, the competition soon adopted two weight divisions: 20 lb and 2 lb. The 2 lb weight class at Critter Crunch was probably the earliest version of the Insect weight class.

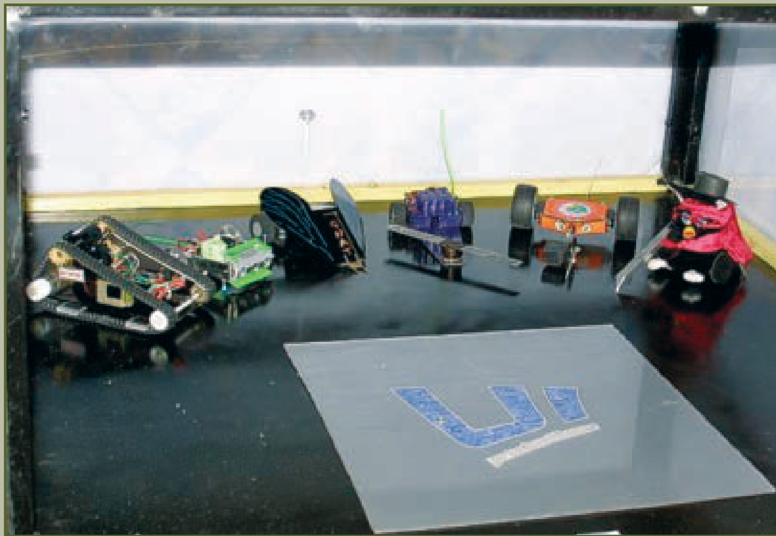
Although Critter Crunch has used this Insect-esque weight class for years, the rest of the nation was slow to catch up. It would take many years before the rest of robot

The 16 invited bots at the BattleAnts competition.

combat learned of this pint sized version of the sport.

In the late 1990s, the UK was experimenting with a variety of Insect weight classes. They eventually settled on the 150 gram "UK Antweight" (in the United States, known as a Fairyweight). United States builders learned of this weight class while competing abroad at the Robot Wars UK competition and on the popular Delphi Forum. A discussion soon began among US builders to adopt a version of the Antweight at home. Andrew Lindsey, one of the founders of NERC (North East Robotics Club) and Delphi Forum frequenter, recalls that "the idea of Antweight bots started with the UK competitors, but there it seemed to be a challenge of how to make the smallest possible robots rather than a low-budget entry class. They were using micro servos and receivers, and trying to figure out what the smallest weight was that still allowed for mobility and some kind of weapon, while the US Antweight class seemed to have been geared more towards low cost and ease of entry."

Because of the differing priorities of the United States builders, they adapted the UK Antweight to an even 1 lb weight class, and the Antweight was (basically) born.


Mystery arena photograph of BattleAnts bots.

According to **BotRank.com**, the first recorded Insect competition took place at Gasworks Park in Seattle, WA on April 14, 2002. It was the first event ever hosted by WAR (Western Allied Robotics). Despite this, YouTube videos exist clearly showing Insect fights in 2000 at an event called CJRC (Central Jersey Robot Conflict) hosted by NERC.

Michael Maudlin's excellent website documents a win by his Antweight TadPole at CJRC '01 and iHXRC '01 (International Hobby and Model Expo) in Chicago. Dr. Maudlin documents these long forgotten events at **www.lazytoad.com/teamtoad/events.html**.

Simultaneously, Sozbots was forming in southern California. Sozbots — which gets its name from an abbreviation of the term "16 oz bots" — held its first event February 21, 2002, according to cofounder Eric Stoliker. Stoliker, along with Patrick Campbell, Brian Roe, and later Peter Abrahamson went on to run many successful Sozbots events.

They also ran an extremely popular Insect parts store online. The Sozbots electronic speed controller is arguably the quintessential Insect component, and became wildly popular among Insect builders in the mid-2000s.

Then — either by coincidence or an act of fate — BattleAnts was held as a demonstration at BattleBots 5.0 in May 2002. It was held outside of the venue for the larger BattleBots, and was an invitation only event for 16 builders. The specifics of this event are a little hazy; the brackets, for example, are nowhere to be found. By one account, the BattleAnts competition used the WAR arena and was hosted by SECR (South East Combat Robotics). In the **photo**, note the UI Productions logo (the company of SECR founding member Mark DeVits).

DeVits, as well as SECR founding member Chris Williamson, were key players in organizing the BattleAnts event. Williamson, like the Sozbots founders, also founded a successful online parts store called **AntBotics.com** which was later sold to the Robot MarketPlace. According to many sources (including a 2004 *SERVO* article), the founding members of Sozbots first came into contact with the Antweight class at this event.

However, a *SERVO* article written about the event in 2002 clearly shows the bots fighting in a Sozbots arena (see next **photo**). Key members of Sozbots even competed at BattleAnts. The previous picture — although featuring many of the same bots that competed at BattleAnts — was probably actually

Excerpt from 2002 *SERVO* article showing Sozbots arena.

from another event. Based on the evidence that I was able to find, it is clear that Antweight events were being held regularly in the northeast by NERC and in southern California by Sozbots before this 5.0 related event.

Despite the conflicting accounts, however, it is certainly clear that BattleAnts was the catalyst that sparked the Insect explosion of 2003; many became enchanted with the pint sized bots at that event, and set out to create their own. The overwhelming majority of the builders I talked to cited BattleAnts as their first taste of the Antweight class.

By 2003, the Insect class had exploded. Events were being held almost weekly all across the country. From the development of the Antweight class out of the UK Antweight fights, to its popularity after BattleBots 5.0, the story of the Antweight class is very much like robot combat as a whole. It was a group effort from a core of builders which sparked the development of the class; it rose to popularity in a matter of months, and is kept active today by a group of devoted builders who are extremely passionate about the little bots.

Come back next month for The Rise of the Insect Part 2: Beetleweights. SV

EVENTS

Upcoming and Completed Events

Upcoming Events for April 2012

RoboGames 9 will be held on April 20th through April 22nd in San Mateo, CA. For further information, please contact **www.robogames.net**.

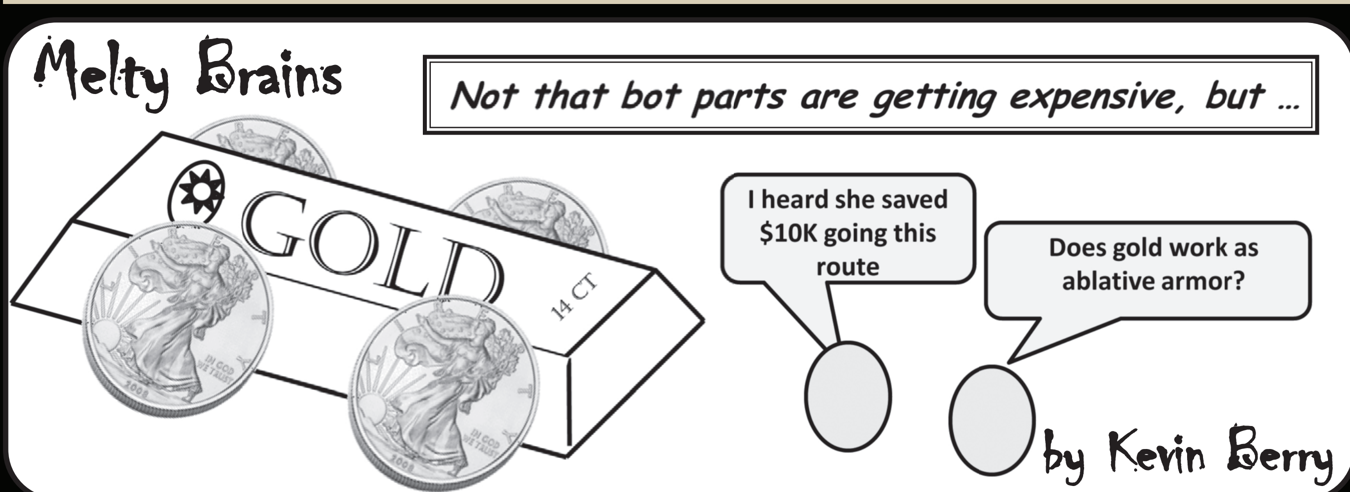
STEM TECH Olympiad 2012 will be presented by the United States Alliance for Technological Literacy in Miami Beach, FL on April 25th through April 29th. For further information, please contact **www.usatl.org**.

Completed Events for February 2012

NWMHobby Expo 2012 was presented by Western Allied Robotics in Monroe, WA on February 11th.

Motorama 2012 was presented by the North East Robotics Club, Inc., in Harrisburg, PA on February 17th through February 19th.

University of South Florida Robotics Interest Group hosted the second annual Robot Interest Group Combat Robot Competition at the USF Engineering Expo on February 18th. SV



Dynamixel and ROS: Towards Affordable Manipulators

Part 1

by Michael Ferguson

Most robot builders have at some point envisioned a human scale robot wandering around their house, cleaning up after them or doing the laundry. Videos of robots such as Willow Garage's PR2 doing just these tasks bring further life to the dream. While most people cannot afford the $400,000 price tag of a PR2, there are some very interesting developments in lower cost robots that can run some of the same software that makes the PR2 so awesome. In this article, we will introduce some of the cool technology that is already out there on the shelves, and in some cases, available for free in open source platforms such as ROS.

The robotics community has seen a quick increase in the capabilities of lower cost robots in the months since the release of the Microsoft Kinect. I think that the new MX-series of Dynamixel servos from Robotis might just be able to spur on a similar increase in the capabilities of low cost ROS-enabled arms.

ROS: The Robot Operating System

The Robot Operating System (ROS) is an open source project started at Willow Garage. The project aims to develop an ecosystem of software that makes robots more useful and easier to build and program.

ROS uses a distributed approach in which developers write many smaller, self-contained programs that carry out specific tasks. Rather than having to compile all of the robot code into a single program, users configure which programs do which parts. This greatly encourages code reuse and because of the open source ecosystem, you can actually download a number of free-to-use programs.

A large number of stable ROS packages already exist. These include things like hardware drivers for Kinect and other Primesense devices (**www.ros.org/wiki/openni_camera**). With new low cost sensors such as the Kinect — which provides both a traditional 2D camera and a 3D camera — robots can now see objects around them and sense the exact location to grab that object.

Higher level packages offer things like speech recognition (**www.ros.org/wiki/pocketsphinx**) or navigation with a Kinect (**www.ros.org/wiki/navigation**). Some up and coming work offers things like collision aware arm planning or object recognition.When writing your own nodes, ROS integrates well with other popular frameworks such as the open source vision framework OpenCV and its 3D counterpart, the Point Cloud Library (PCL). ROS even has bindings to real time robot controllers such as Orocos. Finally, ROS includes a very nice 3D visualization tool called RVIZ (shown in **Figure 1**) which allows users to fuse the visualization of robot state and sensor data; you can finally see exactly what your robot thinks it is seeing. ROS has an excellent set of tutorials which can be found at **www.ros.org/wiki/ROS/Tutorials**.

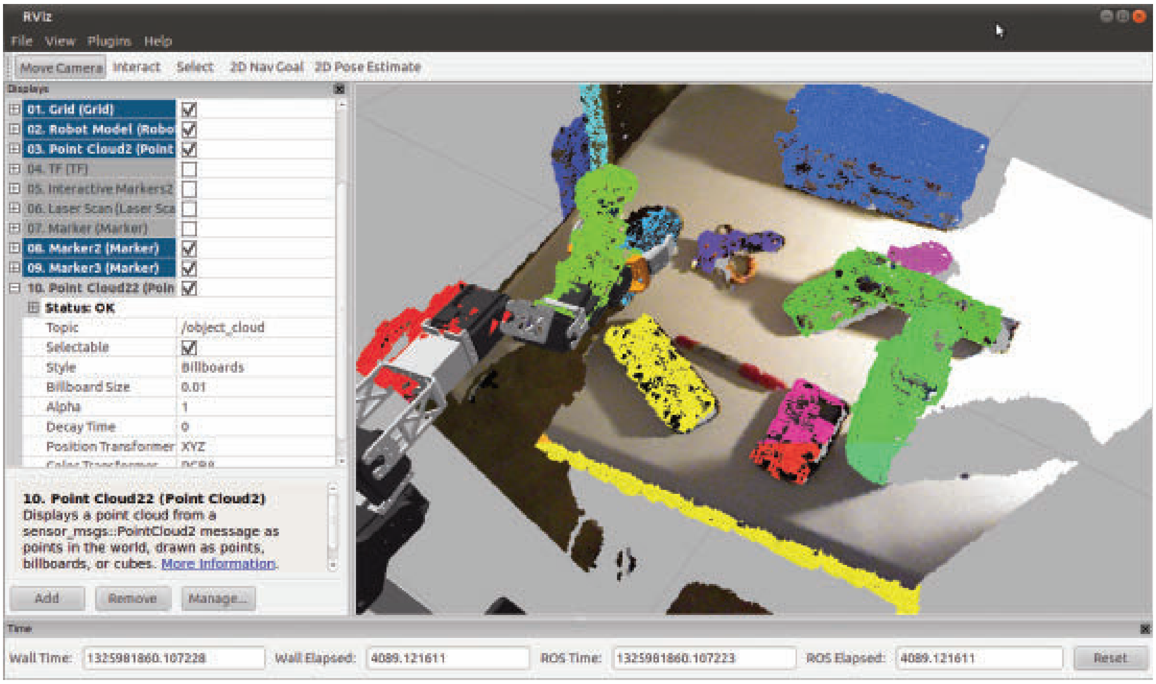


FIGURE 1. A screenshot of RVIZ showing the robot's arm, Kinect point cloud data, and colored results of object segmentation.

Dynamixel Family of Servos

Most people involved in robotics are aware of the Robotis Dynamixel AX-12 servo. This 'smart' servo incorporates a small microcontroller which gives much finer control over the servo than a traditional hobby servo. Dynamixel servos have long been known for their half-duplex TTL serial bus, precise joint position measurement, over-torque and over-temperature shutdown protections, and the ability to tune the torque output.

For years, many have drooled over the higher end RX- and EX-series servos which boast even more torque. Until recently, those higher end servos required an RS-485 bus to communicate, meaning you had to either build your robot entirely out of high end and more expensive servos or you had to have two buses. However, Robotis has recently begun to release their new MX-series of servos (shown in **Figure 2**), starting with the MX-28. The MX series uses the original half-duplex TTL serial bus, making it easier than ever to upgrade a servo when you need more torque. The new brackets released over the past year also have a number of mounting patterns, meaning that most brackets can be connected. You can finally build an arm with a variety of mixed-size servos using only off-the-shelf components.

The MX-series servos improve things in a number of other areas which are essential for building robot arms. The original AX- and RX-series servos had 10-bit resolution over a 300 degree range of motion. This meant that you had 1,024 positions, or a resolution of 0.3 degrees. While 0.3 degrees sounds nice, that can cause a huge displacement at the end of a longer robot arm. The new MX-series servos have 12-bit resolution encoders giving

FIGURE 2. New Dynamixel MX-series servos.

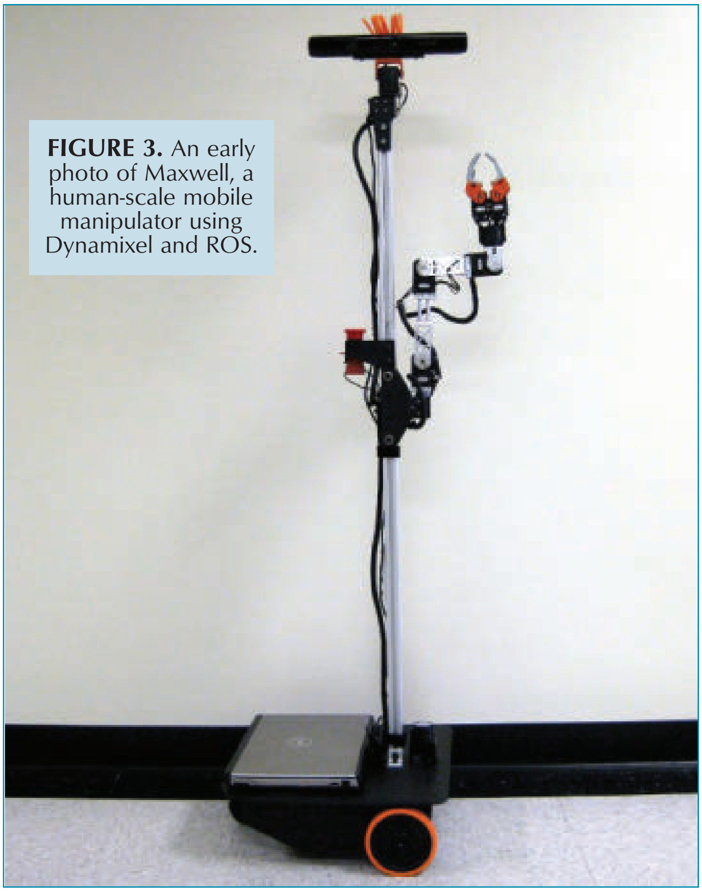


FIGURE 3. An early photo of Maxwell, a human-scale mobile manipulator using Dynamixel and ROS.

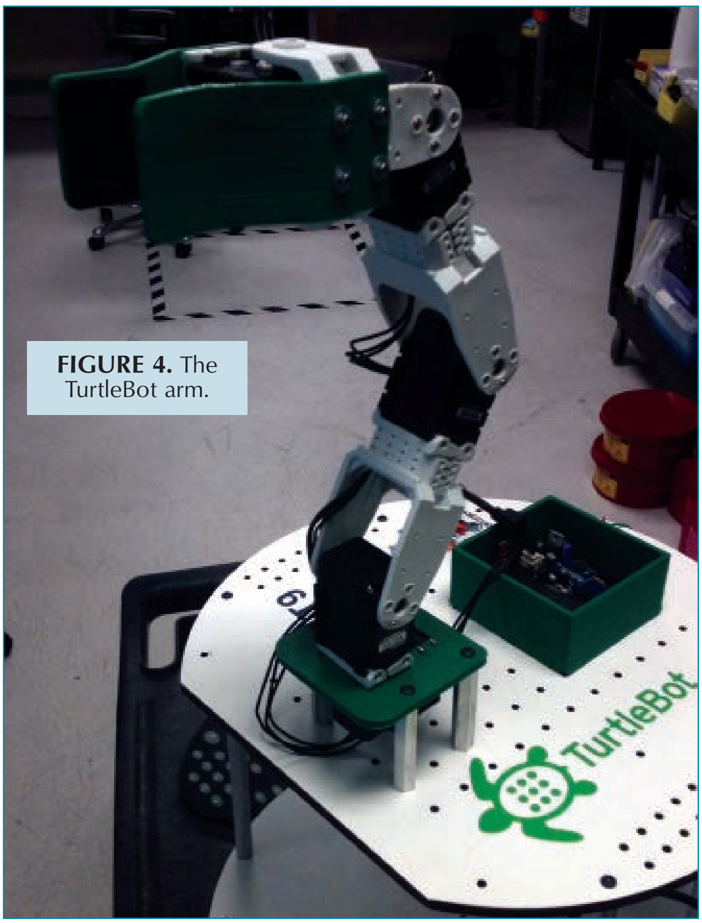


FIGURE 4. The TurtleBot arm.

4,096 positions over 360 degrees of motion, or a resolution of 0.087 degrees. This greatly improves our ability to place the robot gripper exactly where we want it.

The new MX-servos also have a 32-bit microcontroller in them, unlike the older eight-bit controller found in previous series. This allows a maximum bus rate of up to 3 MBps which could be useful if you want to very finely control your robot arm. The improved controller also offers real PID control and access to the individual P, I, and D term gains which opens up a number of cool possibilities for tweaking the servo performance. In all cases, the MX-series servo has mechanical dimensions compatible with its equally-sized RX predecessor. The MX-28 servo is available now; MX-64 and MX-106 servos should be on shelves shortly.

Dynamixel Controllers

Since the Dynamixel servos have a serial bus, you can actually connect them directly to a serial port or to a USB port using a USB-to-serial adapter such as the Robotis USB2Dynamixel. However, we can get even better performance by placing a stand-alone controller in between the USB and Dynamixel buses.

Several years ago (when I helped start the Mech Warfare competition at RoboGames), I found myself wanting an easy-to-use microcontroller board with integrated XBEE wireless radio and the ability to control Dynamixel servos. Out of this need grew the ArbotiX RoboController. After Mech Warfare that year, I found that a number of other people had a similar need. The ArbotiX RoboController is available from Trossen Robotics, and all of the software is released under an open source license.

Why do I bring this up? Because over the past year and a half, I have been using the ArbotiX to connect my robot hardware to a laptop running ROS. I have stopped using the XBEE radio connection (as was used for remote control in Mech Warfare) and started using a direct hard-link wire (a USB-to-serial cable from FTDI) to achieve a more robust connection, and I have added larger motor drivers to support a larger drivetrain for my 25 pound, 5 ft tall mobile manipulator, Maxwell (**Figure 3**). Maxwell is currently undergoing a series of upgrades after which he will have a new arm consisting of mostly MX-64 servos, instead of the current mix of both AX and RX servos which has caused a nightmare of wiring issues. I'll have more details about Maxwell in next month's *SERVO Magazine*.

I built the ArbotiX-ROS drivers from the ground up with the intention that users could quickly configure a robot using only a few lines of a configuration file without writing much (if any) code. The drivers are documented at **www.ros.org/wiki/arbotix** and are released under an open source license. The controllers offer code that allows Dynamixel-based robots to easily leverage the navigation and arm navigation capabilities of ROS.

Some Examples

There are a number of mobile manipulators out there

using Dynamixel servos and ROS. A number of these robots use the ArbotiX. Patrick Goebel (of **pirobot.org**) has used the ArbotiX drivers to power his robot which recently completed a two hour long navigation endurance test. The ArbotiX+ROS combo is also behind the recently released TurtleBot arm (**Figure 4**) — an AX-12 based arm which was originally developed for my Mini-Maxwell robot (**Figure 5**). It uses the iRobot Create and many of the TurtleBot drivers. All of these robots make use of ROS's arm navigation capabilities which allow the arm to move around, avoiding obstacles seen by the Kinect.

Conclusion

This article just barely scraped the surface of what is out there today for people wanting to build a sophisticated, sensor-based robot arm.

Next month, we'll take an in-depth look at the construction of my robot Maxwell, who participated in the RoboGames Symposium and won the 2011 AAAI Small-Scale Manipulation Challenge by playing a number of awesome games of robot vs. robot chess. **SV**

www.servomagazine.com/index.php?/magazine/article/april2012_Ferguson

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

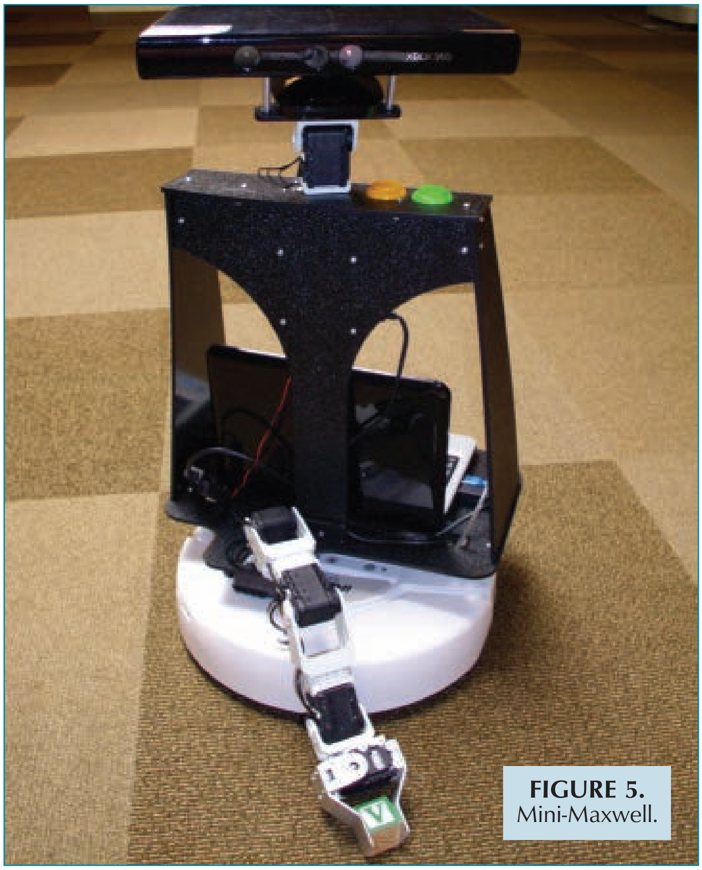


FIGURE 5. Mini-Maxwell.

Creating A SpindleBot

by Robert Doerr

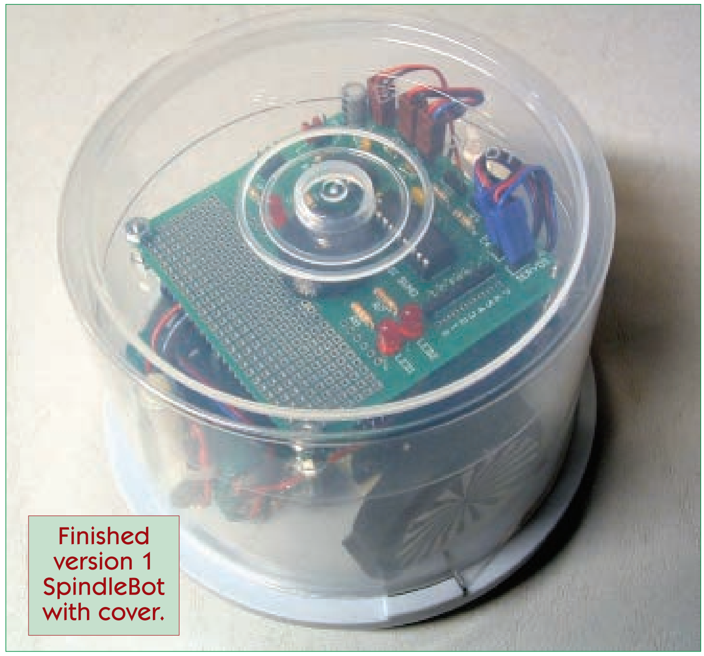
Retro Turtle Robot Construction Tutorial

I may be a bit eccentric, but I spend much of my free time working on personal robots from the '80s like the HERO, Androbot, RB5X, Gemini, and Hubot robots. Even so, I like to explore other areas of the hobby and build other robots too. This project all started when I kept looking at a CD spindle sitting on my desk and thought what a great little robot it could make. It kept reminding me of the early turtle robots. The bottom makes a great base for a robot and with the cover in place, it would have the finished retro look I was after. Besides, it is a great way to recycle something that normally is thrown away. During the construction, many other recycled parts were incorporated into the design. Little did I know that this would end up creating a whole new class of small robots: the SpindleBots!

www.servomagazine.com/index.php?/magazine/article/april2012_Doerr
Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

Birth of the SpindleBot

The first SpindleBot prototype started out when I took a day off to clean up around the house. I already had most of the parts on hand and just needed the time to work on the project. For the body, I selected an empty 50 CD spindle case with a standard twist top since it looked like the ideal size. To keep a neat appearance, everything had to fit within the case so the cover could be installed to finish it off. The trick was to figure out the best way to squeeze everything into such a confined space. For the brains, I used an early BASIC Stamp 1 based Sumo board that needed a home. It was a good board to start out with and it just happened to fit within the CD case. The board was meant to drive a pair of standard RC servos like those found on many small robots. That was ideal since those are the motors I intended to use. One reason these are so popular is that they already have a nice motor/gearbox that can be directly connected to a wheel.

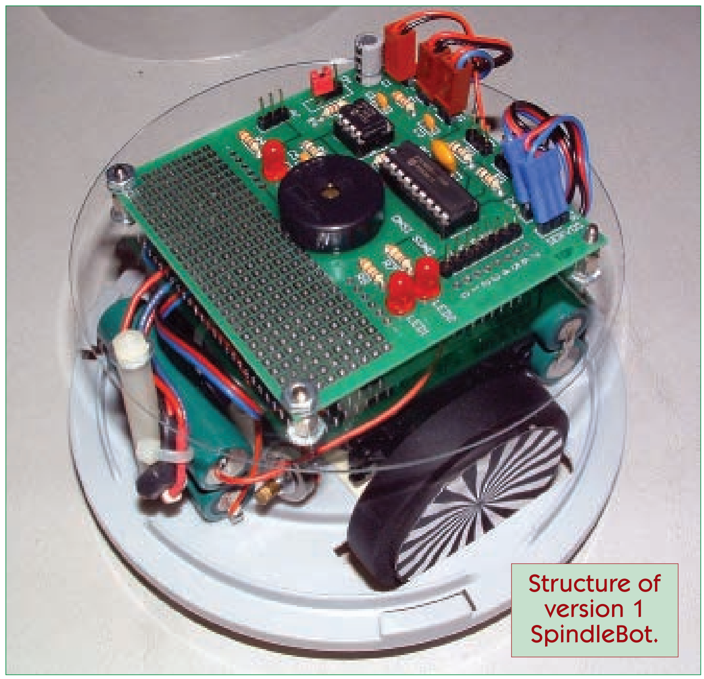


Finished version 1 SpindleBot with cover.

Preparing the Servos

This first version of the SpindleBot used a pair of standard Airtronics servos. Like most servos, these had to be modified for continuous rotation. This is a common modification to make. The first part is to disassemble the gearbox and then trim off the stop on the large output gear. That stop prevents the output shaft from turning all the way around. The second is to pull out the potentiometer that normally fits into the output shaft. Once that is complete, the servo will then be able to continuously turn in either direction.

The potentiometer needs to be adjusted for no servo movement while it receives a pulse approximately 1.750 ms in duration which is a centered servo. I've seen some people cut off the shaft of the potentiometer and leave it in place. Others just remove it and instead add a pair of 2.2K resistors, and have the common lead go to where the adjustment would have gone. However, this method is non-adjustable and with the fixed setting all the calibration has to be done in software.

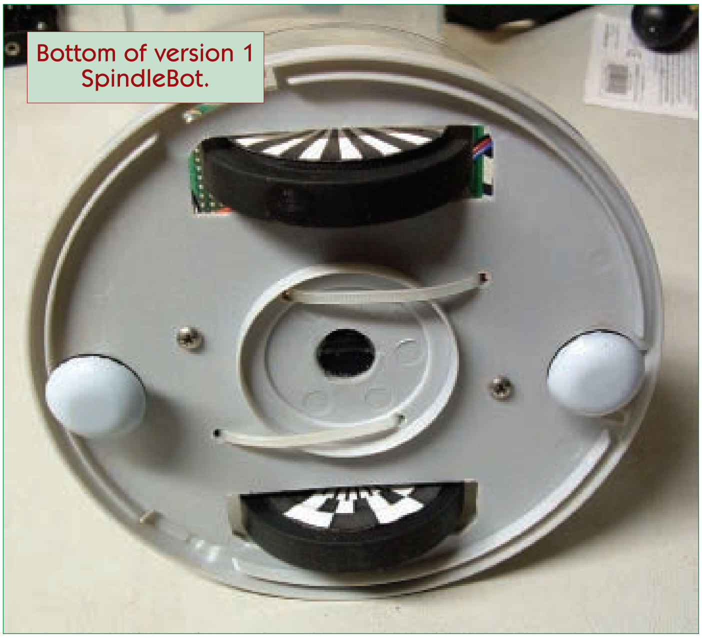


Structure of version 1 SpindleBot.

The method I prefer is to bring the adjustment potentiometer outside of the servo. For many servos like the Airtronics and Hobbico, this is easy since they are usually connected via a small three-conductor ribbon cable. All that needs to be done is to cut down the notch where the servo power/signal comes into the servo, and then route the cable with the potentiometer in the area just notched out under the existing servo wiring. With the potentiometer exposed, the servo can easily be calibrated and adjusted at any time to set the center point. A quick Internet search should find detailed instructions on how to modify almost any other servo.

NOTE: One other servo modification is to remove one of the motors and reverse the polarity so it will run in the opposite direction. This way, the pulse width you send to them means the same for forward and reverse on both

The stop to cut off to modify a servo.



Bottom of version 1 SpindleBot.

Power switch used on SpindleBot.

servos. Otherwise, the pulse widths would mean the opposite from side to side if one motor isn't reversed.

Preparing the Base and Mounting the Wheels

With standard height servos, I had to cut the center spindle off the base in order to make room for mounting. This allowed each of the servos to sit on their side with the bottom of each one nestled up against the other. For the wheels, I used a pair of 2" round rubber feet from an old piece of computer equipment that was being scrapped. Each wheel was glued and then screwed to a standard servo horn. Just about any servo horn will work. The hole in the center allowed me to screw the wheel and servo horn to the servo. That way, it could be removed later if needed.

With the wheels attached, I could then mark the base for the clearance holes to allow the main drive wheels to protrude through the bottom of the base. Working with plastic is relatively easy, and a sharp Exacto or other hobby knife will do the job. There are many different techniques for cutting plastic but I usually make a light cut on the outline and then go over it several more times, cutting a little deeper each time. The motors were then mounted to the base with double-sided tape and a cable tie was placed on each one to keep them secure. On the underside of the base, a small slider was used at the front and rear to help keep the robot level. It did double-duty since the screw holding the slider on also secured the 2" standoffs to hold the tray for the BASIC Stamp board. A power switch and connector for a charger were also added on the bottom of the base to wrap that portion up.

There was just enough room left to install four AA NiMH batteries on the base to keep the center of gravity low. Two were in front of the servos and the other two were behind the servos. These were the tabbed versions with the leads soldered directly to the wires. They connect to the power switch and also to a charger jack through a 10 ohm resistor (poor man's current limiting) which will be used with a 7.5V 200 mA unregulated wall transformer. On top of the two 2" standoffs, I used a plastic CD to create a platform to hold the electronics. Often, CD spindles ship with a completely clear plastic CD which is ideal for this. However, any CD or DVD can be used.

With this tray in place, four short standoffs were used to mount the brains. Running the wires was then a simple task. At this point, it was like any other BASIC Stamp powered robot with a servo drive system. It just had a much more finished look. There is still some room left for future expansion. This little SpindleBot looked okay but I knew it could be improved. It was a good first attempt, but I wanted to make better use of the available area.

Improving the Design – SpindleBot 2.0

The next version (robot #2) was completely redesigned and improved. It is amazing to see the difference between the original SpindleBot and the next version. There is a lot more packed into the same space, and it has a much more polished look and feel.

This new robot really came together. One of the main variations from the first robot is that I wanted to ensure that the center support for the spindle could be preserved.

Sources

Parallax
www.parallax.com

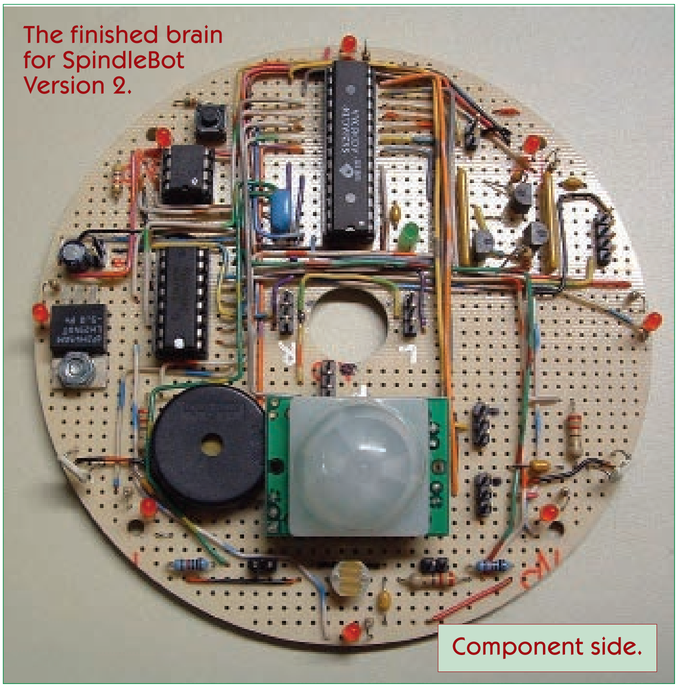
RobotWorkshop
(CDP2 Protoboards)
www.robotworkshop.com

Tower Hobbies
www.towerhobbies.com

RadioShack
www.radioshack.com

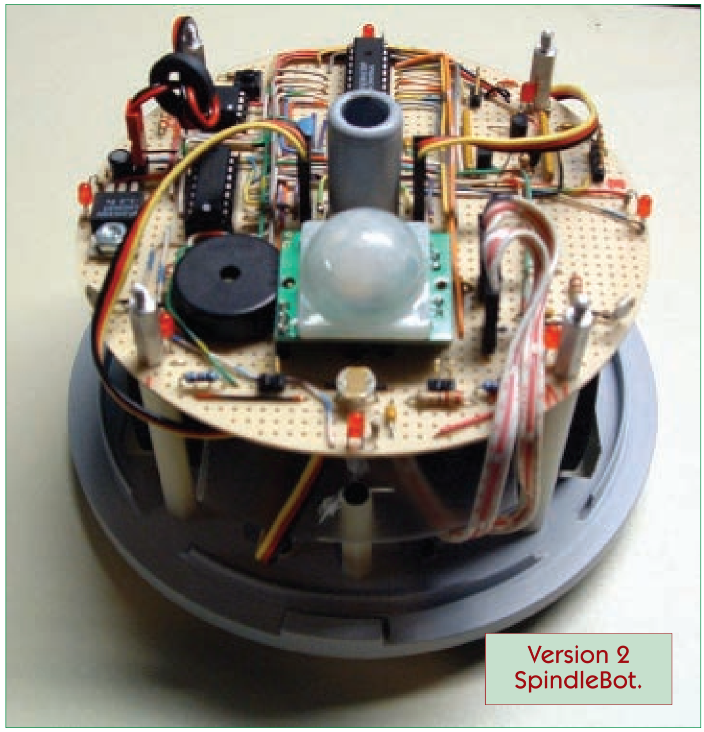
Hobbico
www.hobbico.com

Empty Spindles
(if you can't wait)
www.americal.com/pd/CABX100.html

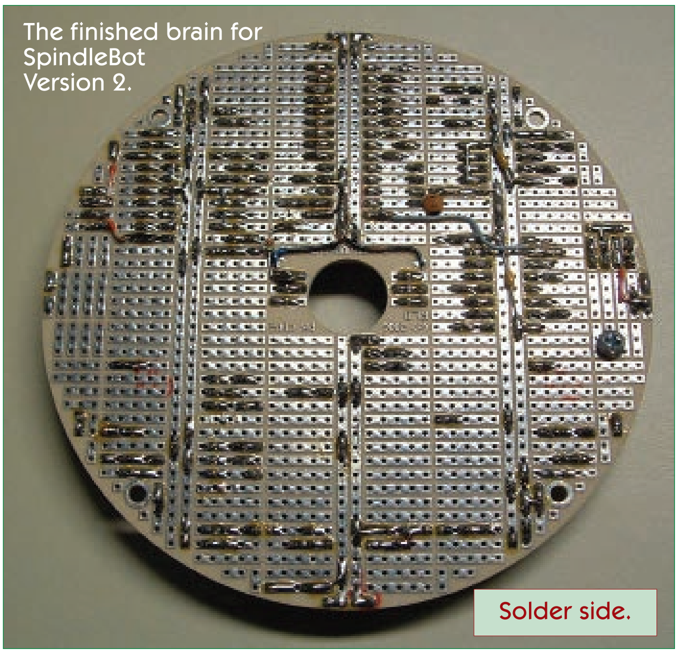


The finished brain for SpindleBot Version 2.

Component side.

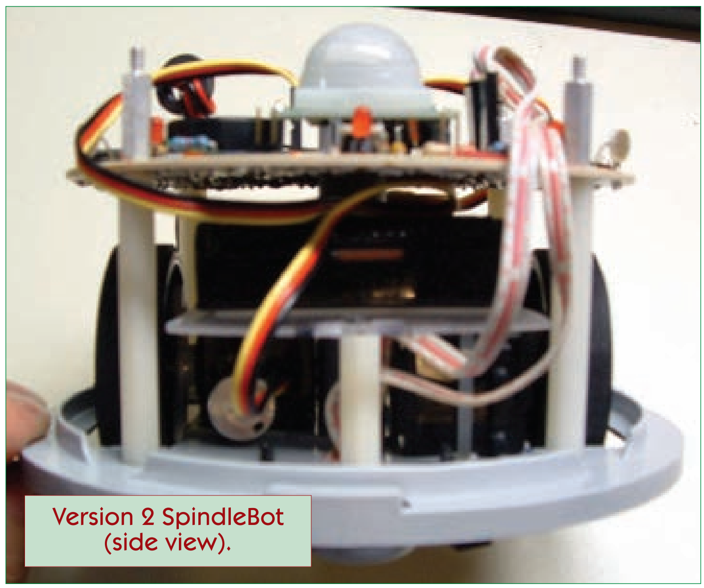


Version 2 SpindleBot.



The finished brain for SpindleBot Version 2.

Solder side.



Version 2 SpindleBot (side view).

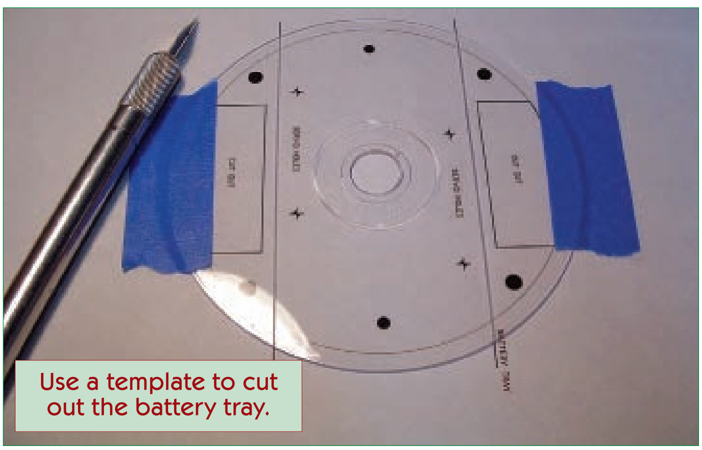
Note the difference between version 1 and version 2 robots.

(The reason this is important will become clear later in the article.) On this robot, the internal structure was built up in layers. The base holds the servo motors, the power switch, and charging connectors. The next level holds the batteries. Like the first robot, the top level will hold the brains of the robot. As the design matured for making a SpindleBot, I was able to create an easy to use template in pdf form that shows where the holes should be drilled for the supports, the cutouts for the wheels, and where to trim down the CD/DVD for the battery tray.

NOTE: During the process of designing and building the SpindleBots, I came across a great source for small robot wheels. They are the large paper feed rollers from HP inkjet printers. They are the perfect size for use in a SpindleBot


Note how to place the template on the base to mark holes.


The bare spindle with the cutouts.


Use a template to cut out the battery tray.

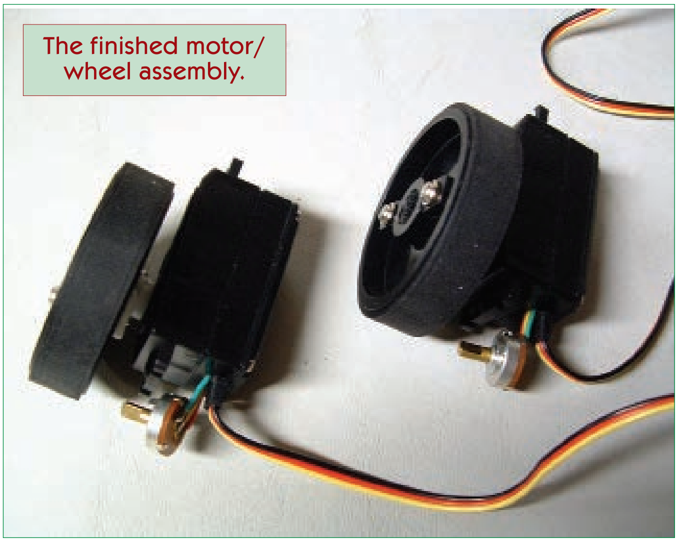

Finished battery tray.


These are from HP inkjets and are used for the robot wheels.

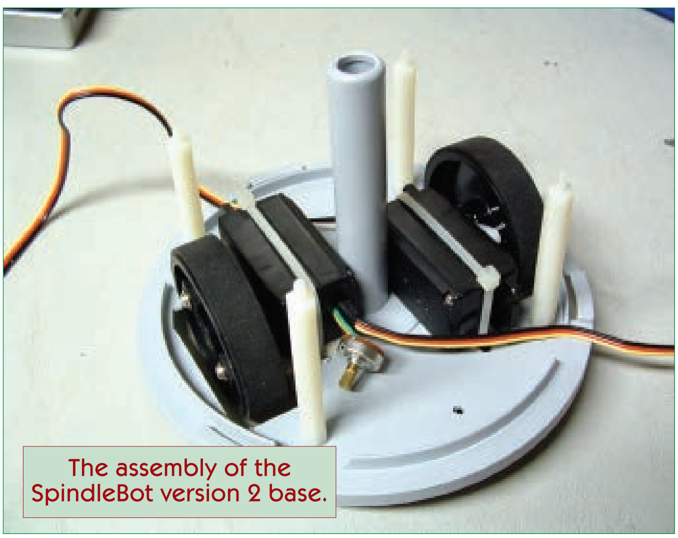

Lots of wheels from recycled printers.

These horn drill gauges help to drill holes in wheels.

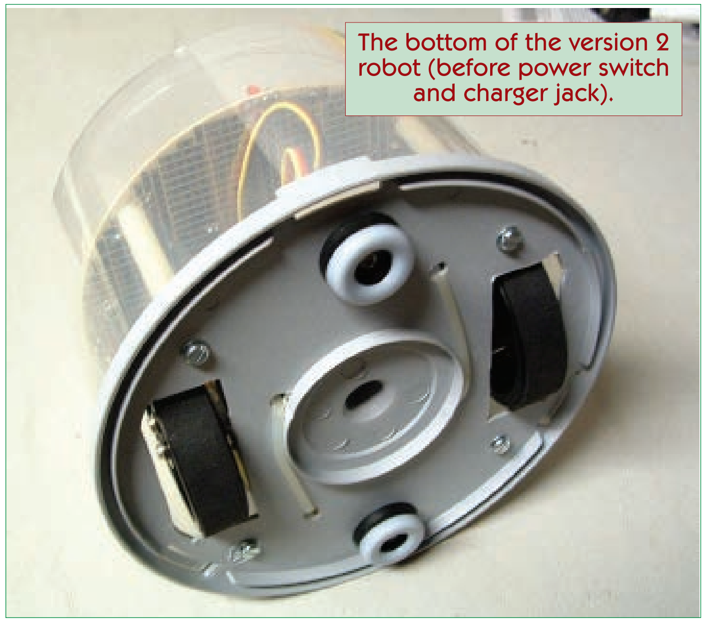
Note how the wheels are attached to the servo horn.



The finished motor/ wheel assembly.



The assembly of the SpindleBot version 2 base.



The bottom of the version 2 robot (before power switch and charger jack).

Low profile servos used on V2 and newer SpindleBots.

and provide excellent traction. There are normally three of these pressed onto the rear shaft in the printer. Removing them isn't hard. The method I use is to open a vise just slightly wider than the shaft, and then use a hammer to drive the shaft out. Once you've liberated the wheels, they are screwed and glued to the servo horn. It is easy to find used inkjet printers very cheap or often free. There are also many other useful parts such as optical sensors, encoders, etc., that can be re-purposed. Pull one of these printers apart and see what you can re-use!

NOTE: To make it easier to mount the wheels to the servo horns, I picked up a pair of Futaba horn drill gauges (part numbers FUTM2400 and FUTM2401) which helps ensure that the spacing for the holes is correct.

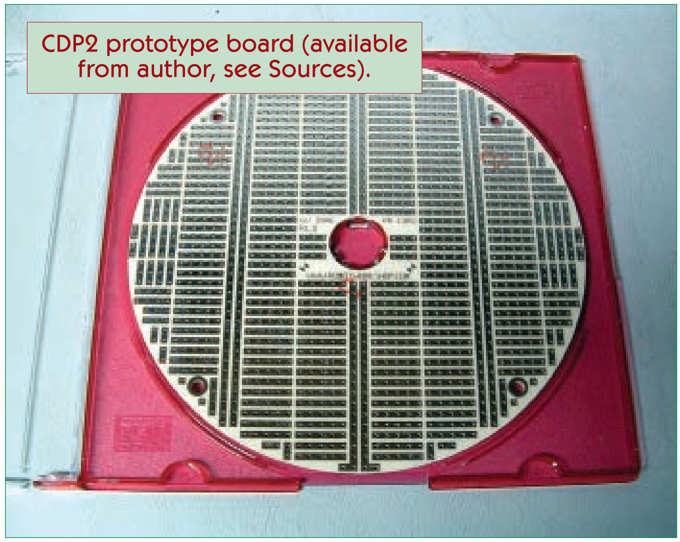
In order to keep the center spindle in place, it wasn't going to be possible to use standard servos. Luckily, there are a variety of servos available. The solution was to use a pair of low-profile Aileron servos for the drive. They are just short enough to provide the clearance needed to keep the center spindle. They are secured to the base with double-sided tape and cable ties like the first robot. For the main drive wheels, I used some of the large feed rollers from an

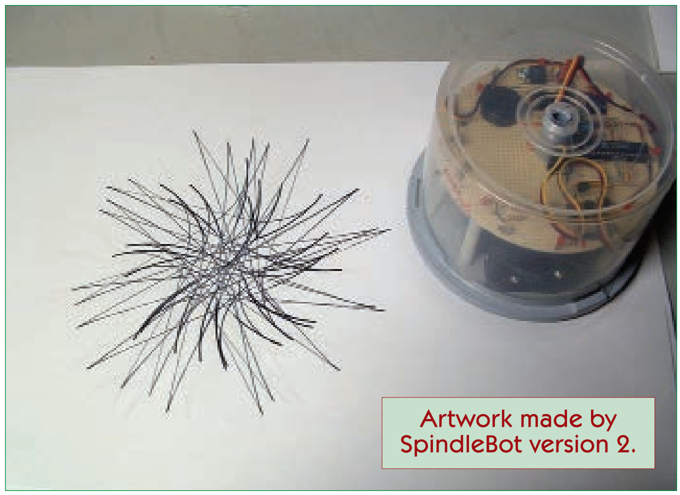
Sliders used at front and rear of SpindleBot base.

Battery holders and charging plug.

old HP inkjet printer. The size is perfect for the SpindleBot and the traction is outstanding. The new wheels are easier to source, as well. The rest of the base is the same with the power switch, charging jack, and sliders all on this level.

A pair of 1" standoffs is used to mount a CD/DVD that has been trimmed down on two sides to clear the wheels. This second level holds the batteries and provides an upgraded 6V supply for the robot. There are two battery holders used. These are attached with double-sided tape. To ensure they sit level, I used a larger drill to counter-sink


CDP2 prototype board (available from author, see Sources).

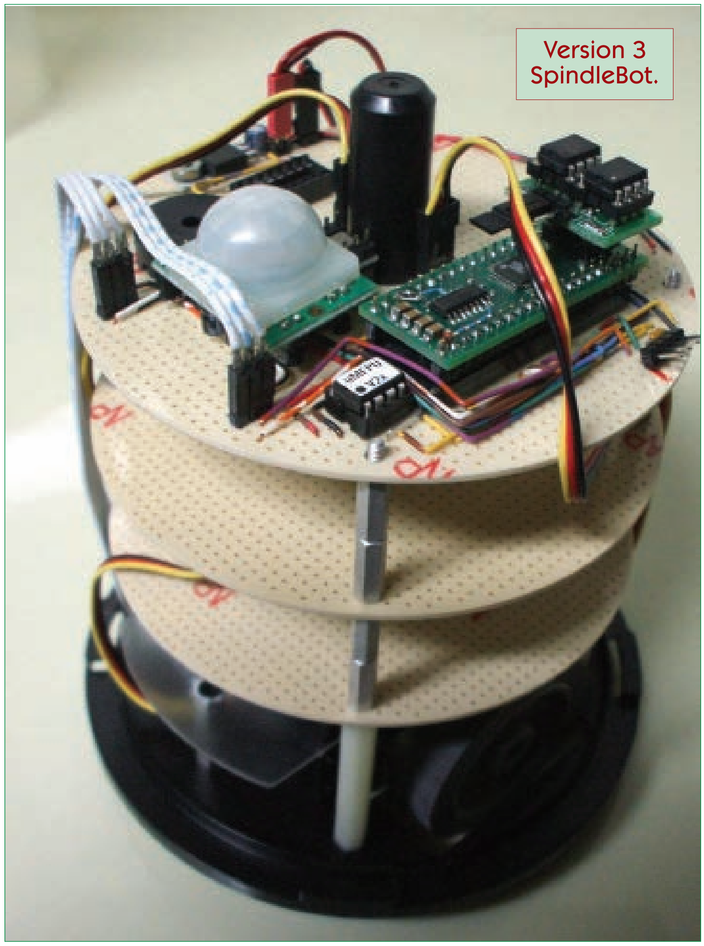

Artwork made by SpindleBot version 2.

each of the two mounting holes. This way, it won't interfere with the screws. The front holds two AA NiMH batteries and the rear one has three AA NiMH batteries. The extra weight towards the back of the robot helps favor the rear slider so it normally ends up touching the ground.

NOTE: In order to make the best use of space for the electronics, a new prototype PCB (printed circuit board) was created to leverage this form factor — the CDPCB. The first board in the series is the CDP2 which is a single-sided PCB that is very easy to work with. With standard .100" hole spacing, it can handle a wide variety of components and microcontrollers. There were times when I needed to isolate sections of the foils underneath. This is a simple process that can be accomplished with either a hobby knife or by lightly using a drill bit that is slightly larger than the width of the pad you want to remove.

The top level holds a CDP2 CD-protoboard which contains all the electronics for the robot. During the construction, I wanted to try using an OEM BASIC Stamp chip. It was all wired by hand on the protoboard using solid phone wire. I've found that wire is very easy to work with, and you can often get lengths of that wire for free. This board uses a standard host of sensors that you'd find on any Stamp style robot like the Boe-Bot or original Scribbler 1

Version 3 SpindleBot (bottom view).



Version 3 SpindleBot.

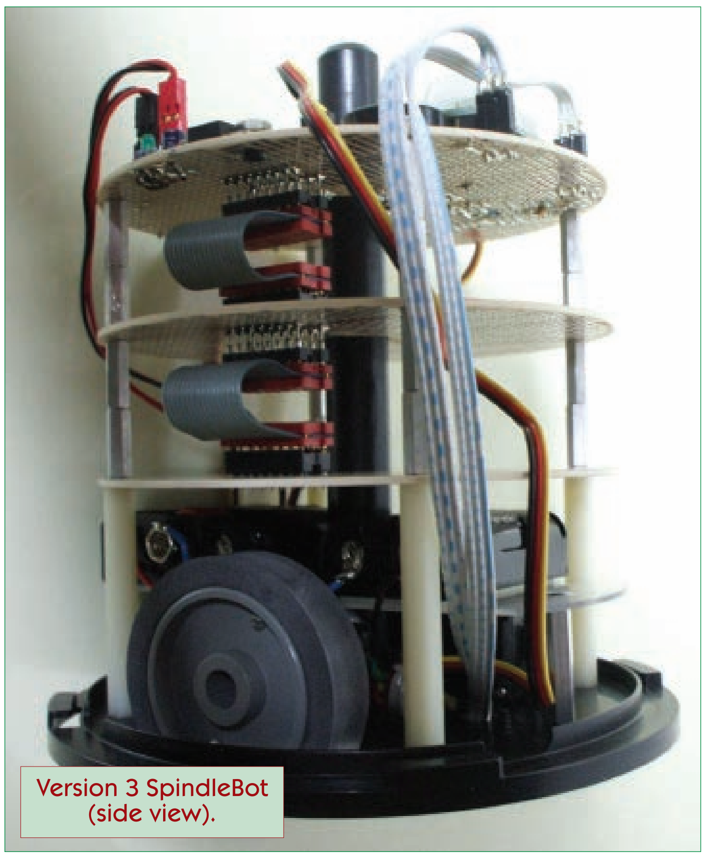
robot. On this SpindleBot, I used a PIR motion sensor and three CDS photoresistors. The outputs are configured to drive two main drive servos, a small piezo speaker, an optional pico servo for pen up/down, and a 74HC595 shift register to control a set of eight LEDs for some cool lighting effects. Most of these are optional, and you can use whatever sensors you want when you build your own version.

The top center of the CD case and the top center of the Spindle were drilled out. This hole allows for a pen to be placed in the center of the robot so it can draw while moving. When drilling into plastic, you need to be very careful not to crack it. To start, just use a small drill to make a pilot hole. Any drill bit 1/8" or less will do. Then, follow up with a Unibit or stepped drill to enlarge the hole. The stepped drills are much less likely to damage the plastic. If you try using a larger drill bit, it will more often than not break the plastic instead of making a proper hole.

When using a 25 or 50 CD spindle, a small pico servo can be added to raise/lower the pen. Since they are so small and the pens are light, it can be attached to the top cover of the CD spindle with double-sided tape. For a more secure mounting, small aluminum angle brackets with screws are an excellent option. An example program that I wrote to test my robot turns the robot into an art bot. It has been a popular demo whenever I have shown off SpindleBots at local schools. I've also tried Scribbler 1 code which worked with some minor modifications to the motor control code and adjustments to pin assignments to match my board.

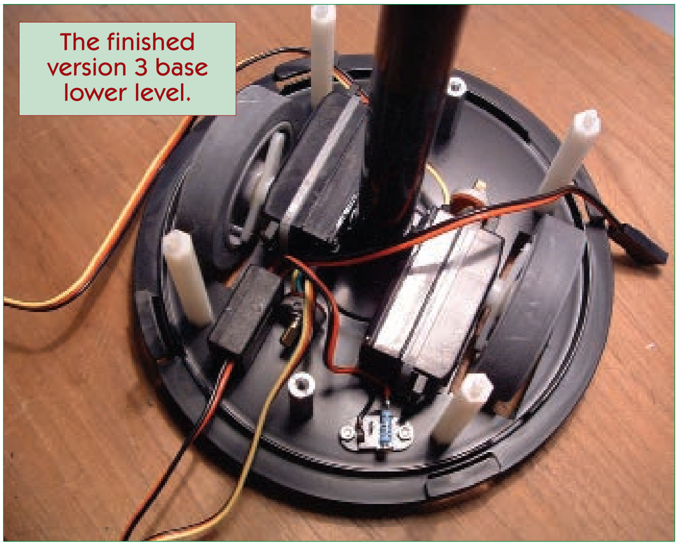
Creating the Ultimate SpindleBot

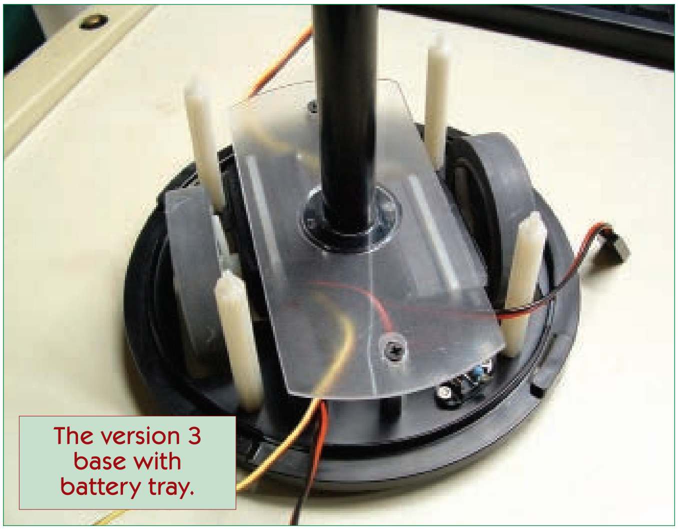
The latest version of the SpindleBot is even more refined. It uses one of the tall 100 CD spindles for its body. The construction is exactly the same as the shorter SpindleBot but has room for more levels of electronics and batteries. The CDPCB boards can be stacked on standoffs to

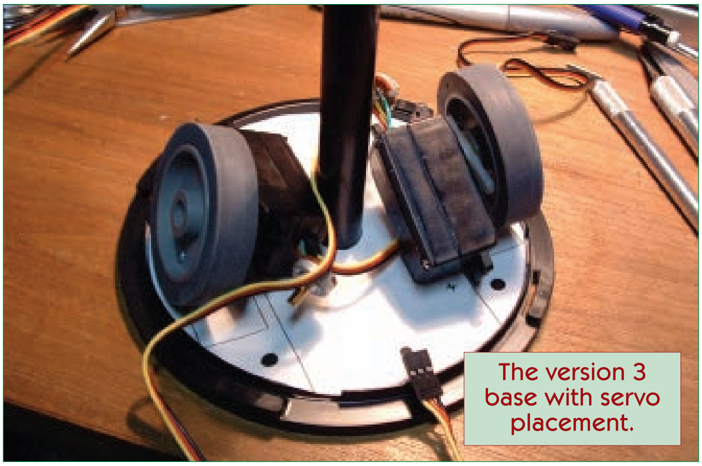


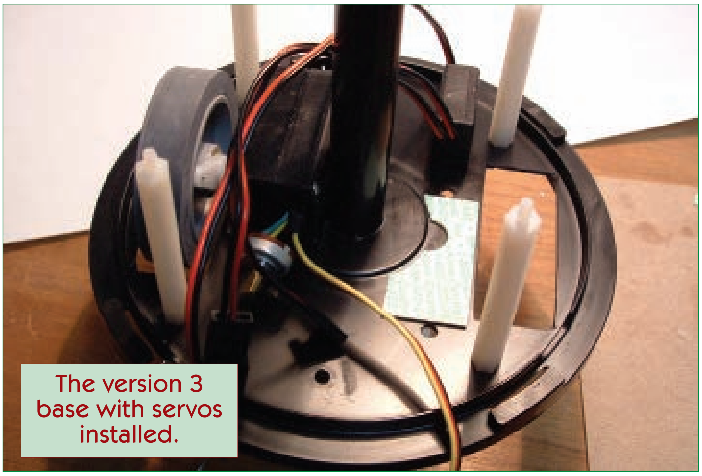
Version 3 SpindleBot (side view).

Battery monitor.


The finished version 3 base lower level.

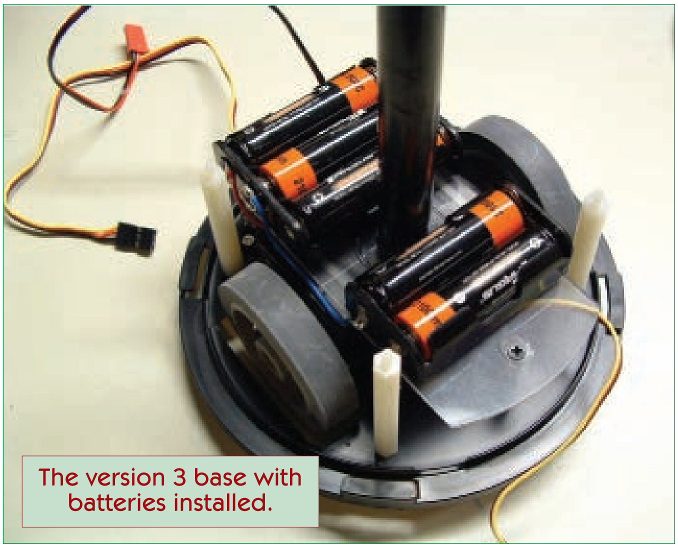

The version 3 base with battery tray.


The version 3 base with servo placement.

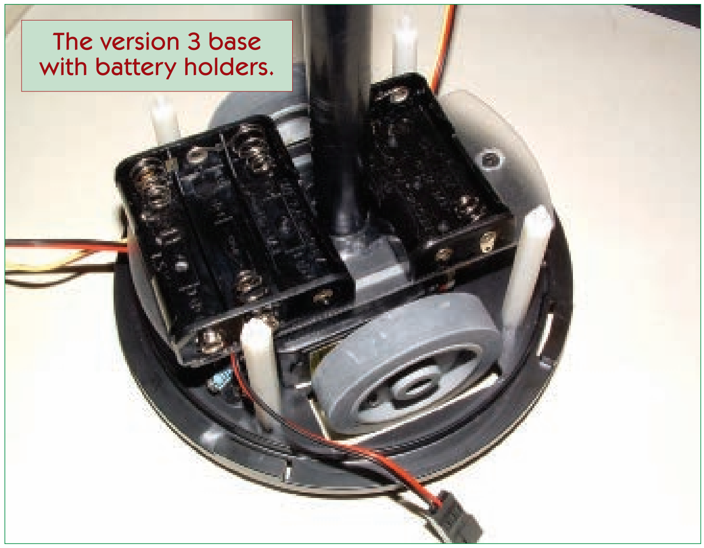

The version 3 base with servos installed.

Parts List

ITEM	DESCRIPTION
1	50 CD (1/2 height) or 100 CD (full height) spindle case
2	Short (Aileron) servos, Hobbico CS-59 or Tower Hobbies TS-59
2	Large paper feed rollers from a recycled HP inkjet printer
2	Small gliders w/spacers
1	Regular CD or DVD to trim for the battery tray
2	1" 4-40 or 6-32 hex spacers w/screws
4	2" 4-40 or 6-32 hex spacers w/screws
1	Dual AA battery holder
1	Triple AA battery holder
1	Controller of your choice (BASIC Stamp, SX48 OEM, Propeller, PICAXE, Arduino, etc.)
2	Parallax line sensors
1	Parallax motion detector
3	CDS sensors
1	Piezo speaker
1	Battery voltage monitor
1	Power switch assembly (Futaba, Airtronics, etc.)
1	Pico servo for optional pen up/down
1	Charging jack and optional current-limiting resistor
1	CDPCB for control logic (more for full height robots)



The version 3 base with batteries installed.



The version 3 base with battery holders.

get at least two more boards on the robot. This version switched to a 40-pin OEM style BASIC 2p40 Stamp, but was upgraded to accept a Propeller processor for even higher performance. Just about any processor could be used if you have one you're more comfortable with.

In order to keep tabs on the battery voltage, I picked up a small voltage monitor meant for RC equipment. It is switchable for either 4.8V or 6V. This can go on the back of the second level and mounts to the battery holder. With it installed, it is easy to tell how the battery is doing and it provides a cool display. It clearly shows the load on the battery increase whenever the servos are moving, and when other circuits are active that draw a noticeable amount of current.

Now with a slick platform, it was time to expand upon that. For line following, a pair of the QTI line sensors were added under the base at the front of each of the SpindleBots. Some of the self-tapping screws recycled from an HP inkjet printer were used to secure the QTI sensors to the underside of the base. There are lots of options for sensors these days. With the tall 100 CD spindle, you can build a robot bristling with a wide array of sensors. One of the next things I plan on adding will be small XBee modules to let all the SpindleBots talk to one another and work as a group.

I haven't tried using a 10 CD/DVD or 25 CD/DVD spindle but I'm sure that someone will make one into an interesting robot. To help classify any new SpindleBots out there, I'm going to put forth the following size guide which is very reminiscent of the old floppy standard: 25 CD/DVD case 1/4 height; 50 CD/DVD case 1/2 height; and 100 CD/DVD case full height. I recently spotted a short 10 CD/DVD spindle case. I guess that could be classified as a 1/8th height spindle.

That's a Wrap

There is now a whole new class of robots that you can build, and I think these will be great in a classroom environment. In a classroom setting, it could help teach an interesting way to recycle and/or re-purpose many parts. Those that can't be used could be identified by the material type (metal, plastic, etc.) and then recycled. It would help teach students about the types of materials used to construct an inkjet printer and how those same materials can be recycled.

If you build your own SpindleBot, I would enjoy seeing a picture of your robot or hearing from you. Whether you are building robots for school or just a hobby, I hope you find building your own SpindleBot as fun and rewarding as I did. SV

Different size spindles to use for making a SpindleBot.

Sounding Off Part 2

Build A Music-Controlled Arduino Tunebot

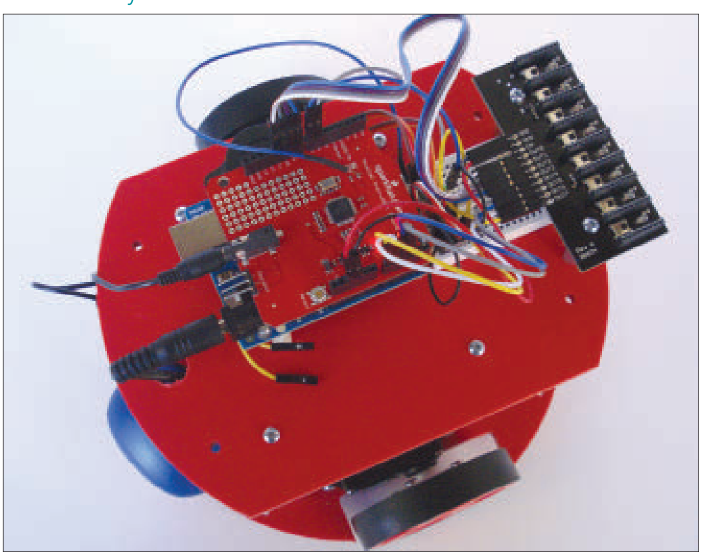
by Gordon McComb

In the classic movie, *Close Encounters of the Third Kind*, the (hopefully) friendly aliens start communicating with us lowly humans with a series of five musical notes. We assume these five notes mean "sure, we come in peace," however, since they never made a sequel to the film, for all we know they were really saying "we prefer earthlings medium rare."

Brain-eating space invaders or not, music has long served as a form of language. Music transcends generations and cultures – most of us would have difficulty reading a book written during the time of James I of England — in Latin, no less. Yet, we can all follow the music of the time. It may not be rock and roll, but you can still dance to harpsichords and lutes. Dig them crazy minstrels!

So, too, your robot can communicate using music. Last month, we looked at producing simple musical tones with just the Arduino and a variety of low cost sound co-processor chips. Trouble is, the music quality of these sources is anything but Hi-Fi. With a low cost synthesizer board, however, you can turn your robot into an orchestra on wheels. Play single notes, chords, sound effects, even musical invitations to dinner with extraterrestrials.

FIGURE 1. The Tunebot is built on an ArdBot platform, or most any Arduino-based robot will do.



Discuss this article in the *SERVO Magazine* forums at **http://forum.servomagazine.com**.

In this installment of Sounding Off, you'll learn how to add a repertoire of realistic sounding instruments to your robot by using MIDI. Though I'll concentrate on programming the MIDI sound using an Arduino, the same techniques apply to any microcontroller you'd like to use. I'll call the musically gifted robot described here as Tunebot; see **Figure 1** for what it looks like.

Before we begin, last month I said we'd cover both MP3 and MIDI sound making. Alas, the finished article was way too long, so I'll concentrate on MIDI sound this time, and save MP3 for the next installment. Fair enough? Good! Let's begin.

Much Ado About MIDI

MIDI stands for Musical Instrument Digital Interface. It's a standard method for controlling electronic instruments – it does other jobs, too, but music is what we're interested in here. The MIDI specification covers the data transmission itself, the electrical connection, even the hardware used to link everything together.

For a self-contained robot, we're mainly interested in the data talking part. MIDI speaks by sending short messages over an asynchronous serial connection.

- The data sender is referred to as a controller. A common MIDI controller is an electronic keyboard but there are many other kinds; for the Tunebot, the controller is an Arduino.
- The data receiver is referred to by various names such as sound module, synthesizer, or sound bank. Its job is to listen for commands sent by the controller and turn them into musical notes. The sound module is connected to an amplifier and speaker so you can hear the music.

Most messages are only two or three bytes long. Each message starts with an eight-bit command (or status) byte, followed by one or more seven-bit data (or parameter) bytes. The combination of command and data byte(s) of a single message is an event. **Figure 2** shows a simplified example of a three-byte MIDI message. By starting every message with an eight-bit byte and then only using seven-bit data bytes, the sound module can more easily keep sync between itself and the controller.

As a programming practice, command bytes are usually entered in code in hexadecimal (hex) format. On the Arduino, hex values are preceded with an 0x prefix. For example, 0xB0 is the same as decimal 176. Data bytes are entered as decimal or hex values — whichever method you prefer. I'll provide some examples shortly.

Of Instruments, Banks, and Channels

The most basic MIDI module is programmed to reproduce the sound of at least one instrument, but many are capable of synthesizing numerous instruments: acoustic and electric piano; guitars; various woodwind and brass instruments; drums; bongos; cymbals; and much more.

Sound modules have built-in defaults, and for instruments the default is usually an acoustic grand piano. To change the instrument so the module plays something else, you send the "change instrument command," 0xC0, followed by the number of the instrument you want to use. Instrument numbers span from 0 to 127, with 0 being the default.

For instance, to change to instrument #12 – whatever it might be – you simply transmit a couple of bytes from the controller to the MIDI sound module:

```
0xC0 12
```

(Before going further, take note that the documentation for the sound module probably lists the instruments starting at 1, but programmatically you specify them beginning with 0. Just keep that in mind when you meant to choose "Voice Oohs" and instead get "Choir Aahs.")

Recall that seven-bit data bytes are limited to just 128 values. Many modern MIDI modules are capable of reproducing more than 128 instruments, including all the various drum and special effects sounds. The number of instruments can be increased by separating them into two or more banks. With up to 128 banks and 128 instruments per bank, you could conceivably choose from 16,384 different instruments. There may be few MIDI modules with that many instruments programmed into them, but the potential is there.

As shown in **Figure 3**, instruments are played through channels; each channel can sound a different instrument. MIDI supports 16 channels. Many (but not all) command bytes are defined as channel messages. The eight bits of the command byte are divided into two four-bit sections (see **Figure 4**):

- The first four bits specify the command, such as changing instruments.
- The second four bits indicate the MIDI channel, from 0 to 15 (16 channels total).

For example, the command byte for switching instruments (0xC0) begins with the binary bits 1100. Channel_0 is binary 0000, making the whole command byte:

```
1100 0000
```

Other variations are:

```
11000001   Channel_1
11000010   Channel_2
11000011   Channel_3
```

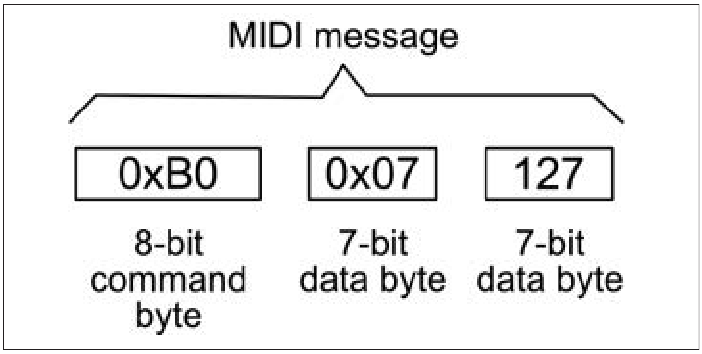


FIGURE 2. MIDI messages are composed of just a few bytes, an eight-bit command byte (it always has its eighth bit set to 1), and one or more seven-bit data bytes.

and so on. To combine a command byte with a channel number, simply add them together. Popular programming practice performs this function using binary OR, which on the Arduino is done using the | (pipe) operator:

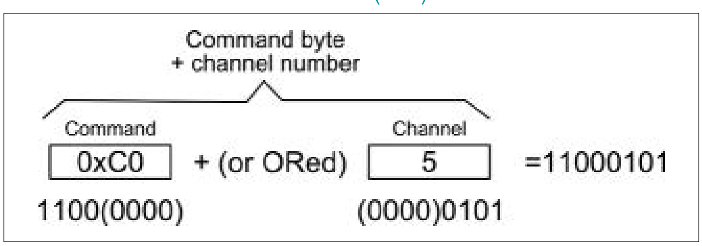
```
0xC0|3     // Bits: 11000011
```

Note: As with instruments, channel numbers in MIDI are typically noted as 1-16, but programmatically are referred to as 0-15. So in the list above, Channel_0 is actually MIDI channel 1, Channel_1 is MIDI channel 2, and so on. This difference really only matters if you're interfacing with other MIDI equipment — like a MIDI keyboard — where channels are numbered 1 through 16. For programming using an Arduino and a MIDI breakout board like the one in Tunebot, the channel offset numbers make little or no difference.

Understanding Channel Voice Messages

Switching instruments to play something else is an example of modifying the channel voice. Channel voice messages are those that alter the sound on a specific channel, rather than for the entire sound module. There are seven channel message commands in common use:

FIGURE 4. Channel voice bytes are constructed by adding or ORing the four most significant bits (MSB) of a command byte with the four least significant bits (LSB) of the channel number.



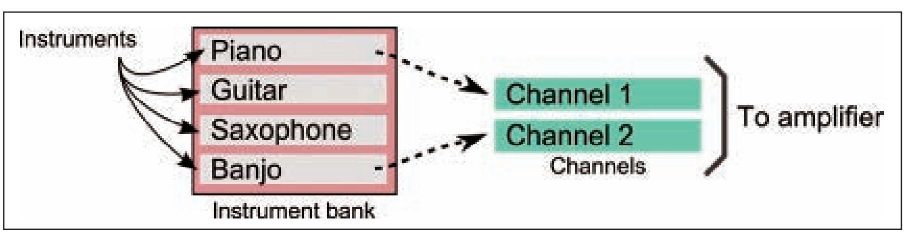


FIGURE 3. MIDI devices synthesize one or more instruments that are kept in a repository called a bank. Instruments are played through any of 16 channels.

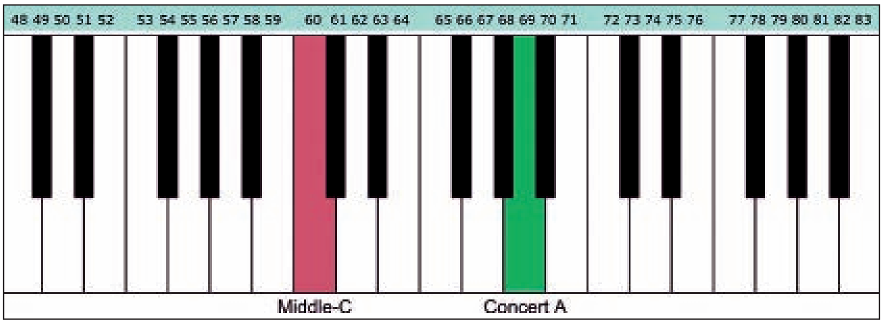


FIGURE 5. MIDI pitches are defined using a standard numbering system that corresponds to the white and black keys on a keyboard.

Of these, the following four are used in Tunebot:

- *Program Change*: Selects which instrument to play.
- *Note On*: Turns on a note of a specific pitch and velocity (more about these terms below).
- *Note Off*: Turns off the note.
- *Control Change*: Selects a variety of operations, such as changing the volume of the channel or selecting a different bank of instruments.

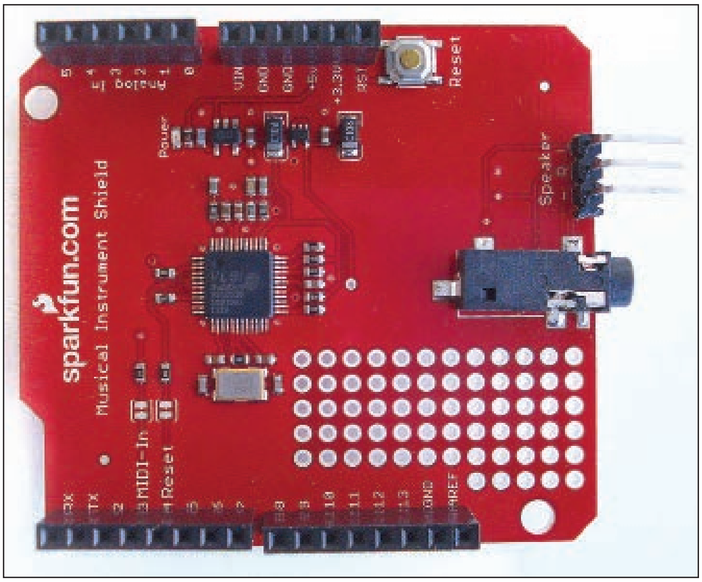


FIGURE 6. The Music(al) Instrument Shield from SparkFun is a low cost MIDI sound module that plugs directly on top of the Arduino.

Command	**Binary***	**Hexadecimal**
Note Off	1000*nnnn*	0x80
Note On	1001*nnnn*	0x90
Polyphonic Key Pressure	1010*nnnn*	0xA0
Control Change	1011*nnnn*	0xB0
Program Change	1100*nnnn*	0xC0
Channel Pressure	1101*nnnn*	0xD0
Pitch Bend Change	1110*nnnn*	0xE0

* The *nnnn* bits specify the channel (0 to 15).

Each note you play on a MIDI device has two primary characteristics: pitch and velocity. Pitch is the frequency of the note, and velocity is its volume. On a piano, each white or black key on the keyboard delivers a different pitch. How hard and quickly you press down on the key determines its volume, or velocity.

MIDI uses a simple numbering system to denote pitch; these numbers (see **Figure 5**) match the white and black keys on a piano. Given a standard 88 key piano keyboard, the MIDI pitch numbers vary from 21 (low tone) to 108 (high tone). The sequential numbering counts both the white and black keys. Very low and very high tones are possible by extending the pitch values through the full 0-127 range. For reference, middle C on the piano is MIDI pitch value 60. It has a pitch (sound frequency) of 261.6 Hz (Hertz, cycles per second).

For note velocity, the values range from 0 (silent) to 127 (full volume). There is also a separate channel volume command (see below) that sets the overall level for the whole channel. Values likewise range from 0 to 127 for overall volume. The example sketches in this article demonstrate setting both note velocity and channel volume.

An example Note On message might look like this:

```
0x90|1 69 100
```

- 0x90 is the Note On command byte; 1 is added (ORed) to it for channel 1.
- 69 is the pitch of the note. The value 69 represents the A above middle C.
- 100 is the note velocity. A value of 100 is about 3/4 full volume (full = 127).

Note that I'm showing only the command byte as hexadecimal; the rest are plain ol' decimal.

Understanding Control Change Messages

The control change message allows for an extended range of channel functions. These messages use the 0xB0 command byte with a channel number, a control byte, and a data byte to

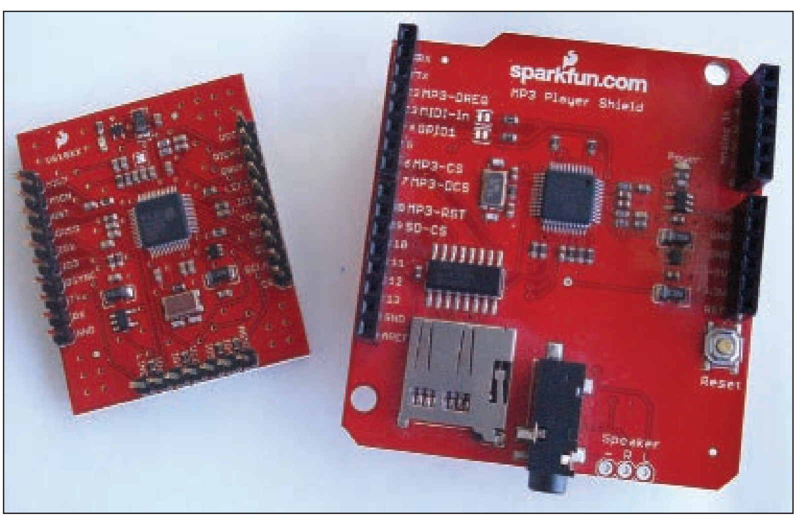


FIGURE 7. Alternative MIDI sound modules include these two board-level products, also from SparkFun: a universal breakout board for use with a microcontroller; and a combo MIDI/MP3 shield.

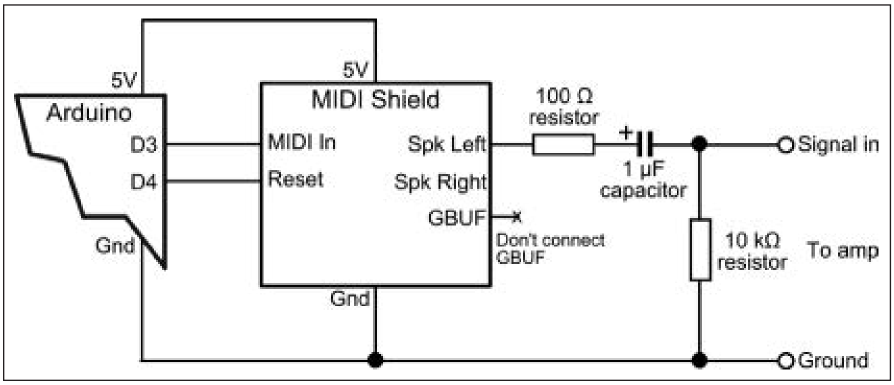


FIGURE 8. Recommended hookup for connecting the output of any VLSI VS10xx-based MIDI sound module with an external amplifier. This is from the manufacturer's datasheet. Only one stereo side is shown connected.

set the value. The three most common control change messages – and the ones used in the example sketches – are:

Option	Control Byte
Bank Select	0x00
Channel Volume	0x07
All Notes Off	0x7B

Examples:

```
0xB0|0   0x00   0x78    // Sets bank 0x78 (decimal
                        // 120) on channel 0
0xB0|5   0x07   63      // Sets volume to 63 on
                        // channel 5
0xB0|3   0x7B   0       // Turns off all notes
                        // playing on channel 3
```

Connecting a MIDI Sound Module to the Arduino

Thanks to the "maker movement" there are now a number of affordable MIDI sound modules available for the Arduino and other microcontrollers. **Figure 6** shows a MIDI sound module shield from SparkFun, designed for the Arduino Uno (and compatible) development board.

Variations include shields that combine MIDI with MP3 playback, as well as breakout boards that provide additional connection pins for more advanced hookups. **Figure 7** shows both of these styles (also from SparkFun). The MIDI/MP3 shield includes an onboard micro-SD card reader that provides storage for the sound files.

Despite all the pins on these boards, most modules require only a couple of connections for MIDI operation:

- MIDI input. This pin accepts asynchronous serial at a specific baud rate of 31250 bits per second. Use the Arduino SoftwareSerial object library to transmit serial data directly to the board.
- Reset. This pin provides a method of resetting the MIDI chip on the sound board to a known state. Do this after the board has been powered up, but before sending any MIDI data. Resetting is accomplished by momentarily bringing the pin LOW, then HIGH again.

Figure 8 shows the basic wiring points between the Arduino and MIDI board. If you're using a shield, the

LISTING 1 – MIDI_VerySimple.ino

```
#include <SoftwareSerial.h>
SoftwareSerial MIDI(255, 3);    // Rx, Tx (Rx not used, Tx pin=3)

void setup() {
  MIDI.begin(31250);            // Set up serial comm to MIDI
  resetMidi(4);                 // Reset MIDI device (pin 8)
  ctrlMIDI(0, 0x07, 127);       // Set channel volume (0-127) to full
  ctrlMIDI(0, 0x00, 0x79);      // Select 'melodic' bank
  sendMIDI(0xC0|0, 0, 0);       // Select instrument in bank
}

void loop() {
  noteOn(0, 60, 127);           // Turn on middle-C note, velocity=127
  delay(1000);                  // Wait one second
  noteOff(0, 60,  127);         // Turn off note
  delay(1000);
}

// Turn note on (press key)
void noteOn(byte channel, byte note, byte attackVelo) {
  sendMIDI( (0x90 | channel), note, attackVelo);
}

// Turn note off (release key)
void noteOff(byte channel, byte note, byte releaseVelo) {
  sendMIDI( (0x80 | channel), note, releaseVelo);
}

// Set controller value with channel
void ctrlMIDI(byte channel, byte data1, byte data2) {
  sendMIDI( (0xB0 | channel), data1, data2);
}

// Send bytes to MIDI device
void sendMIDI(byte cmd, byte data1, byte data2) {
  MIDI.write(cmd);
  MIDI.write(data1);
  // Pass only if valid 2nd byte
  if( (cmd & 0xF0) <= 0xB0 || (cmd & 0xF0) == 0xE0 )
    MIDI.write(data2);
}

// Reset MIDI device
void resetMidi(int resetPin) {
  pinMode(resetPin, OUTPUT);
  digitalWrite(resetPin, LOW);
  delay(100);
  digitalWrite(resetPin, HIGH);
  delay(100);
```

LISTING 2 – MIDI_Simple.ino

```
// Channel voice messages (all are 8-bit, 128 and above)
#define    NOTE_OFF                        0x80    // 2 bytes: key, velocity
#define    NOTE_ON                         0x90    // 2 bytes: key, velocity
#define    CONTROLLER                      0xB0    // 2 bytes: controller num, value

// Control change messages
#define    CTRL_BANK                       0x00    // Select bank of instruments
#define    CTRL_VOLUME                     0x07    // Set volune (0-127)
#define    CTRL_INSTRUMENT                 0xC0    // Set instrument within bank
#define    CTRL_ALL_OFF                    0x7B    // All notes off

// Instrument banks
#define    BANK_MELODIC                    0x79    // "Melody" instruments
#define    BANK_PERCUSSION                 0x78    // Percussion (drum) instruments

// General values
#define    Instrument                      5       // Instrument: electric piano 2
#define    NOTE_C4                         60      // Middle-C on piano
#define    NOTE_A4                         69      // A above middle-C (concert pitch A)
#define    HIGH_AGOGO                      66      // High agogo (use with percussion)
#define    CHAN_1                          0       // Channel (1 of 16)

// Pin assignments
#define    midiIn                          3       // MIDI input
#define    resetPin                        4       // Reset

byte m_velocity = 127;                             // Note velocity (volume)

#include <SoftwareSerial.h>
SoftwareSerial MIDI(255, midiIn);                  // Rx, Tx (Rx not used)

void setup() {
  MIDI.begin(31250);                               // Set up serial comm to MIDI
  resetMidi();                                     // Reset MIDI device
}

void loop() {
  ctrlMIDI(CHAN_1, CTRL_VOLUME, m_velocity);           // Set channel volume
  ctrlMIDI(CHAN_1, CTRL_BANK, BANK_MELODIC);           // Select channel bank
  sendMIDI(CTRL_INSTRUMENT, Instrument, 0);            // Select instrument in bank
  noteOn(CHAN_1, NOTE_C4, m_velocity);                 // Turn on note
  delay(1000);
  noteOff(CHAN_1, NOTE_C4,  m_velocity);               // Turn off note
  delay(1000);
}

// -- Standard functions --
// Turn note on (press key)
void noteOn(byte channel, byte note, byte attackVelo) {
  sendMIDI( (NOTE_ON | channel), note, attackVelo);
}

// Turn note off (release key)
void noteOff(byte channel, byte note, byte releaseVelo) {
  sendMIDI( (NOTE_OFF | channel), note, releaseVelo);
}

// Set controller value with channel
void ctrlMIDI(byte channel, byte data1, byte data2) {
  sendMIDI( (CONTROLLER | channel), data1, data2);
}

// Send bytes to MIDI device
void sendMIDI(byte cmd, byte data1, byte data2) {
  MIDI.write(cmd);
  MIDI.write(data1);
  // Pass only if valid 2nd byte
  if( (cmd & 0xF0) <= 0xB0 || (cmd & 0xF0) == 0xE0 )
    MIDI.write(data2);
}

// Reset MIDI device
void resetMidi() {
  pinMode(resetPin, OUTPUT);
  digitalWrite(resetPin, LOW);
  delay(100);
  digitalWrite(resetPin, HIGH);
  delay(100);
}
```

connections are made for you and no additional wiring is needed. Should the shield use different pins than those shown, you'll need to alter the pin designations in the program sketch.

Making Tunes

Listing 1 demonstrates basic functionality of making MIDI sounds from the Arduino. The sketch — which is designed for Arduino 1.0 — begins with including the SoftwareSerial object library (it comes with the Arduino IDE). The library creates an object named MIDI which is set to transmit data on pin D3.

The setup() function starts the MIDI serial link at 31250 baud, then resets the board by toggling the reset pin LOW, then HIGH.

The remainder of the setup function sends a series of messages to the MIDI board. These messages are transmitted with the assistance of some helper functions — ctrlMIDI and sendMIDI — found later in the sketch.

- ctrlMIDI is for passing control change messages. The function takes three arguments: the channel number, the specific control to change, and the value to change it to. Example: The line ctrlMIDI(0, 0x07, 127) alters the volume on channel 0 to *127*.
- sendMIDI is for passing generic messages. This function also takes three arguments: the command byte, and up to two data bytes. The

Choosing a MIDI Sound Module

Planning on extending your Arduino sound making into MP3? Then, you may want to opt for the SparkFun MP3 Player Shield (DEV-10628). The MP3 board is very similar to the MIDI-only Music Instrument Shield, but it also exposes pins for controlling MP3 playback.

If you use the MP3 player shield, you need to re-arrange some of the wiring for the servos and the Parallax line following module. As the pin definitions need to be different, you must change the sketches accordingly. See **www.robotoid.com** for alternative versions of all the sketches for use with the SparkFun MP3 player shield.

statement line sendMIDI(0xC0|0, 0, 0) sets the instrument on channel 0 to *0*.

All of the boards from SparkFun use the VLSI VS10xx series of MIDI/MP3 chips and in these, instrument 0 is an acoustic grand piano. The VS10xx chips offer two instrument banks, referenced as 0x79 for a set of "melodic" instruments, and 0x78 for a set of drum and other percussion. If you don't specify a bank, the chip will default to bank 0 which is the same as the melodic bank.

Take a closer look at the sendMIDI function itself. Notice that there's a bit of code to determine if the message contains two or three bytes:

```
if( (cmd & 0xF0) <=
0xB0 || (cmd & 0xF0)
== 0xE0 )
```

The logic is this: If the most significant four bits – the ones on the left – are equal to or less than 0xB0, *or* equal to 0xE0, the message is assumed to have three bytes. If the test doesn't meet these requirements, the message is assumed to have just two bytes. The sendMIDI function takes three bytes regardless, so when passing a two-byte message just fill in the third parameter with a 0.

LISTING 3 – MIDI_Chords.ino

```
#include <SoftwareSerial.h>
#include "instruments.h"
#include "constants.h"

byte m_bank = BANK_MELODIC;        // Bank select (melodic, percussion)
byte m_instrument = 0;             // Instrument IN bank (melodic only)
byte m_note = 0;                   // Note (key, pitch)
byte m_velocity = 127;             // Note velocity (volume)

byte minKey = NOTE_C4;             // Beginning pitch for demo
byte maxKey = NOTE_C4 + 12;        // Ending 1 octave higher

// Starting and ending instruments
byte minInstrument = Acoustic_Grand_Piano ;
byte maxInstrument = Rock_Organ ;

SoftwareSerial MIDI(255, midiIn);  // Rx, Tx (Rx not used)

void setup() {
  Serial.begin(9600);
  MIDI.begin(31250);               // Set up serial comm to MIDI
  resetMidi();                     // Reset MIDI device
}

void loop() {
  playBasic();                     // Continuous loop
}

// Main routine, plays block of pitches (notes)
//   within a selected instrument range
void playBasic() {

  ctrlMIDI(CHAN_1, CTRL_VOLUME, m_velocity);     // Set channel volume
  ctrlMIDI(CHAN_1, CTRL_BANK, m_bank);           // Select channel bank

   // Iterate through instrument(s) to play
  for(m_instrument = minInstrument ;
   m_instrument <= maxInstrument ;
   m_instrument++) {

    // Select instrument within bank
    sendMIDI(CTRL_INSTRUMENT | CHAN_1, m_instrument, 0);

    // Play through notes
    for (m_note = minKey ; m_note <= maxKey ; m_note++) {
      noteOn(CHAN_1, m_note, m_velocity);
      noteOn(CHAN_1, m_note+4, m_velocity);
      noteOn(CHAN_1, m_note+7, m_velocity);
      delay(300);

      //noteOff(CHAN_1, m_note,  m_velocity);    // Turn off just 1 note
      ctrlMIDI(CHAN_1, CTRL_ALL_OFF, 0);         // Turn off channel sounds
      delay(100);
    }
    delay(150);                           // Pause before next instrument
  }
}

// Remainder of Standard functions code
```

The loop() function cycles through, turning a note on and off at one second intervals. Two function calls are used here: noteOn and noteOff. They're simply wrappers to the sendMIDI function, and make the coding simpler. Both take the same arguments, but they do different things:

- noteOn turns on a note of a specific pitch and at a specific velocity (loudness).
- noteOff turns the note off. You want a noteOff for every

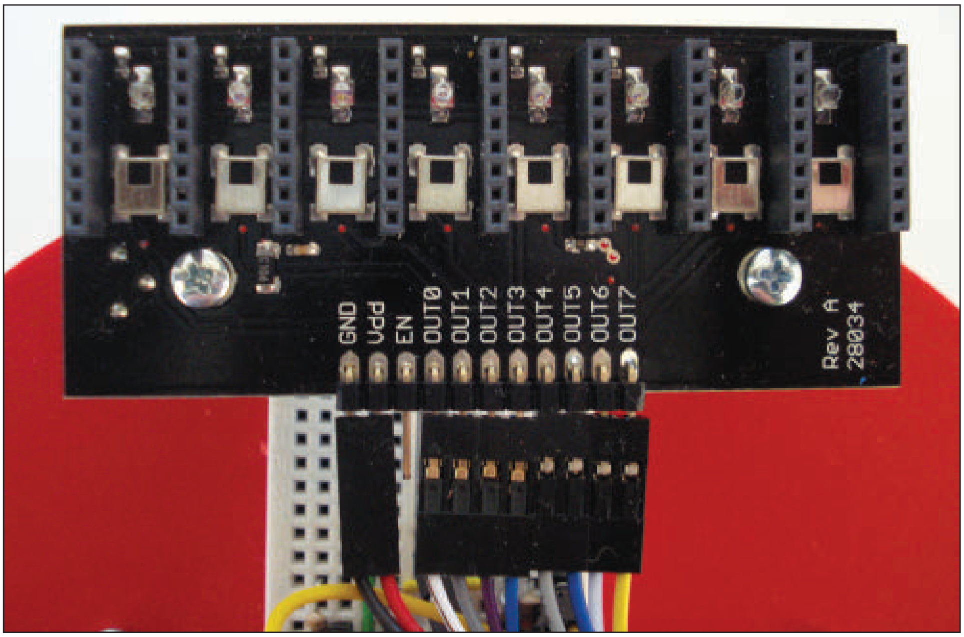


FIGURE 9. The Parallax line follower module contains eight modulated infrared emitters and IR receivers. Tunebot uses it instead as a piano keyboard — playing the keyboard operates the robot.

noteOn, or otherwise the notes will just congeal into one another. (This continues until the MIDI chip runs out of the internal resource space.)

The first argument of the noteOn/noteOff methods is the channel. I'm using Channel 0, but there's nothing stopping you from using another channel. The second argument is the pitch; in this case, it's middle C on the piano. The third argument is the velocity. I'm using the highest value possible: 127.

Using Self-Descriptive Constants

Programming MIDI is made easier by using constants instead of raw hexadecimal or decimal values. Rather than trying to remember that 0xC0 switches instruments, you can use a constant like CTRL_INSTRUMENT which mnemonically helps you to remember what the value does.

Constants are defined in an Arduino sketch using either of two methods:

```
const int SOME_VALUE = 5;
```

or

```
#define SOME_VALUE 5
```

Both do the same thing: They define a symbol that evaluates to a number. When the sketch is compiled, the Arduino substitutes each instance of SOME_VALUE with the number 5. Constants are similar to variables in that both let you reference a value by name. Unlike variables, however, constant definitions don't take up memory in the Arduino. Take notice that when using the #define statement, there's no semi-colon at the end of the line.

Listing 2 demonstrates the same functionality as the previous sketch, but uses constants to refer to the command and data bytes. While the sketch is longer, it takes up the same amount of program space inside the wee innards of the Arduino.

Playing Multiple Notes

The VS10xx chips are polyphonic, meaning they can play more than one note at a time (conversely, monophonic means one note at a time). You can play multiple notes on the same channel or on a different channel. If the notes are from the same instrument, it's easier to use the same channel. Simply repeat the noteOn statement for each pitch you wish to play.

Conversely, if you wish to play notes from different instruments, you'll need to specify an instrument on each channel. Then, call noteOn for each note, referencing the channel you wish to use.

Listing 3 shows how to use multiple notes to create chords. The sketch also shows cycling through a series of

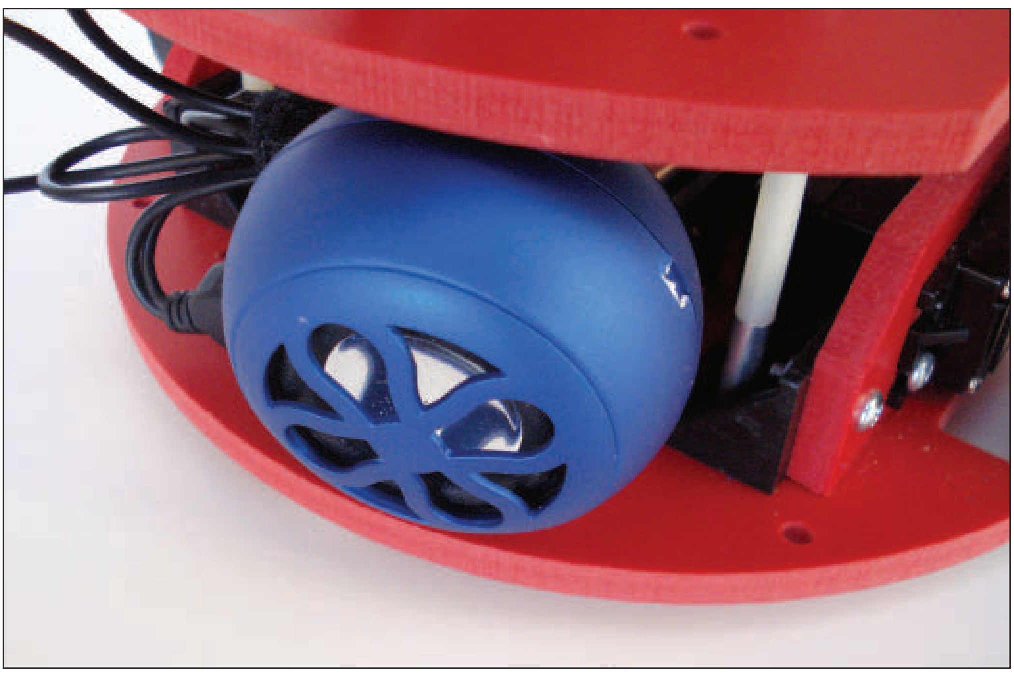


FIGURE 10. A lightweight capsule speaker is connected to the MIDI shield by way of the interface circuit in Figure 8.

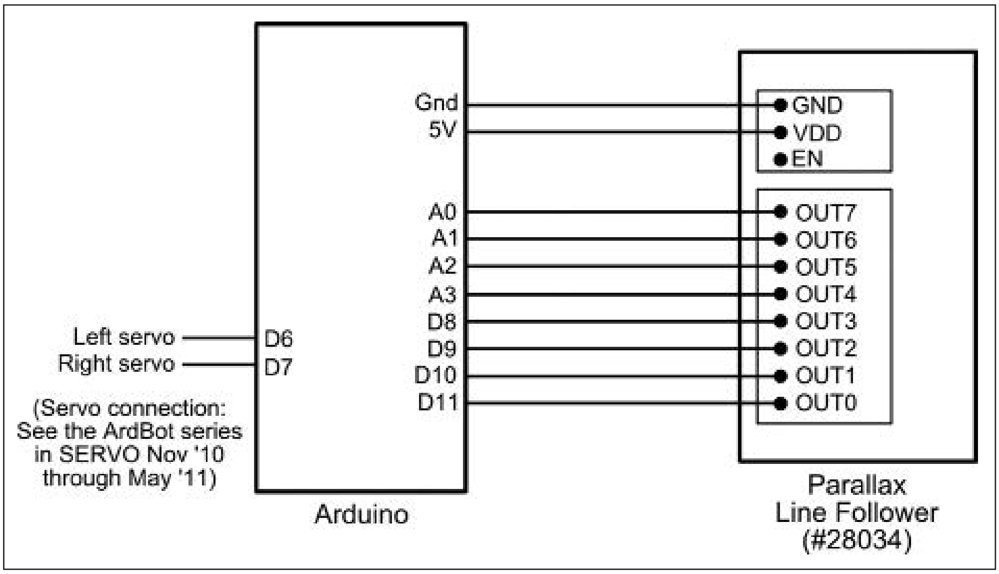


FIGURE 11. Wiring diagram between the Arduino and the Parallax line follower module. The servos are connected to Arduino pins D6 and D7. See the December '10 issue of *SERVO* for details on how the ArdBot's servos are connected.

instruments, playing a full octave's worth (12 steps) for each instrument. You can use this program to sample the various instruments to discover how they sound.

Like the MIDI_Simple sketch, the program in **Listing 3** also uses constants to define numeric values. Only this time they're placed in separate header files to make it easier to manage and update the code. These header files are kept in the same folder as the sketch, and are referenced at the top of the sketch using #include statements.

*Important! The **Listing 3** sketch will not compile without the extra header files.* They're included in the zip file with all the sketches for this article at the download link. Check the URL at the beginning of this article.

The multiple notes are played using this code:

```
noteOn(CHAN_1, m_note, m_velocity);
noteOn(CHAN_1, m_note+4, m_velocity);
noteOn(CHAN_1, m_note+7, m_velocity);
```

The m_note variable contains the root note to play The additional two notes are defined as m_note+4 and m_note+7; these numeric intervals create a major chord. (Remember: The MIDI pitch values count half steps; that is, both the white and the black keys.) The trio of notes are contained in a for loop which cycles through a series of 12 half step notes.

For every note that's turned on, you need to eventually turn it off. This can be done by simply repeating the pitch with a noteOff statement, or you can use a special "All Notes Off" command to remove all notes playing on the channel. The MIDI_Chords sketch in **Listing 3** uses the latter method to demonstrate how it's done.

Tip: Play around with the delay after the notes are turned on. Some instruments require a longer attack — the amount of time for the note to come to full volume, timbre, and form. If the delay is too short, the notes will literally sound cut off.

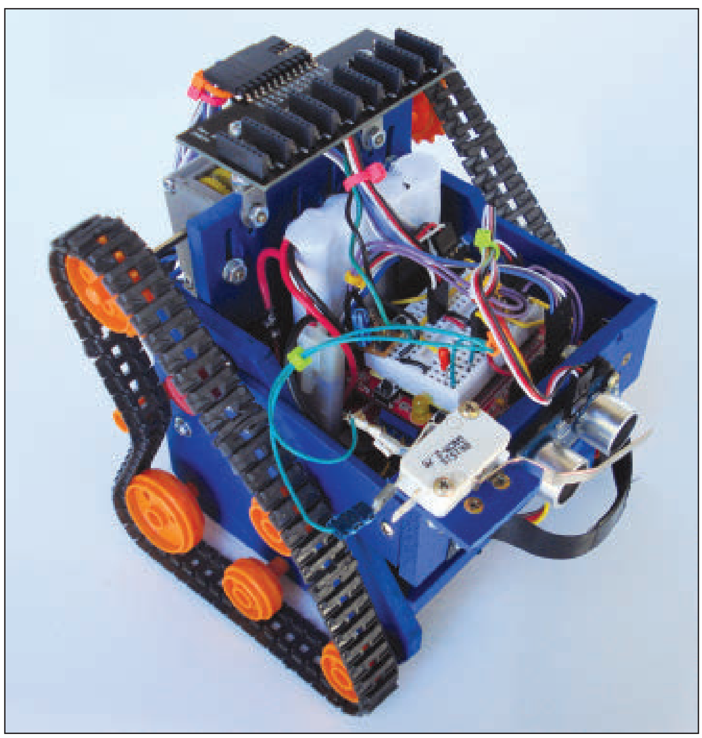
Building and Using the Tunebot

With the basics of MIDI out of the way, we can now turn to the Tunebot (shown in **Figure 1**). I'm using an ArdBot base which is detailed in *SERVO* November '10 through May '11. A kit of body parts is available at Budget Robotics (see the **Sources** box for more info) or you can cut out your own. The ArdBot is a low cost expandable robot base that uses the Arduino Uno as its central brain.

Complementing the Arduino attached to the top of the ArdBot is a SparkFun MIDI shield (DEV-10587), an amplified "capsule" speaker, and a Parallax eight-sensor line follower module (#28034). The file follower module attaches to the front of the ArdBot on one inch standoffs (included with the module) and faces up, as shown in **Figure 9**. By using your fingers or hand, you "play" tunes on the Tunebot to control its behavior. Notes you play operate the servo motors and make sounds through the MIDI board and speaker.

Note the resistor and capacitor interface between the MIDI board and amplified speaker in **Figure 8**. This interface is recommended by the manufacturer, and prevents overloading the MIDI shield or amp. Specifically, don't hook up the GBUF pin (shown on the speaker connections) as a ground; the GBUF pin is *not* at ground

FIGURE 12. Alternative Tunebot design, using a pair of Tamiya three-speed gear motors and rubber tracks.



voltage. Though the MIDI output is stereo, I'm showing it using only one output channel.

Use a small self-powered amplifier, like the capsule type in **Figure 10**. These amps contain their own rechargeable battery. Connection to the Parallax line follower module is shown in **Figure 11**. The line follower module contains an onboard sensitivity adjustment; set it to give you the greatest range. See the documentation for the module for additional information.

The Tunebot uses a sketch called MIDI_Piano. The program listing is too lengthy to reproduce here; you can download it at the article link.

To use the Tunebot, upload its sketch to the Arduino, plug in power to the servos, and set the robot on the ground. Simple movement patterns over the line follower module control the bot:

- Pass your fingers over the center sensors to make the robot go forward. Activate the center sensors again and the robot reverses direction.
- Activate the right-most sensors to turn right; the left-most sensors to turn left.
- Cover all the sensors (or at least the first and last) to make the robot stop.

Experiment with the line sensor coding to add other finger gesture controls. For example, you might add code that listens for a specific three or four note tune in order to activate the Tunebot. Create some code to play your favorite songs while the Tunebot is moving.

Add sound effects – use instruments from the drum bank – and sound them off in response to other sensors you've attached to the robot. How about a cymbal crash when a bumper switch is activated. Or, maybe the slide of a trombone to indicate distance from an ultrasonic rangefinder. **Figure 12** shows an alternative version of the Tunebot, with several sensors mounted on the front to detect nearby objects.

Adding Digitized Sounds, Music, and Effects

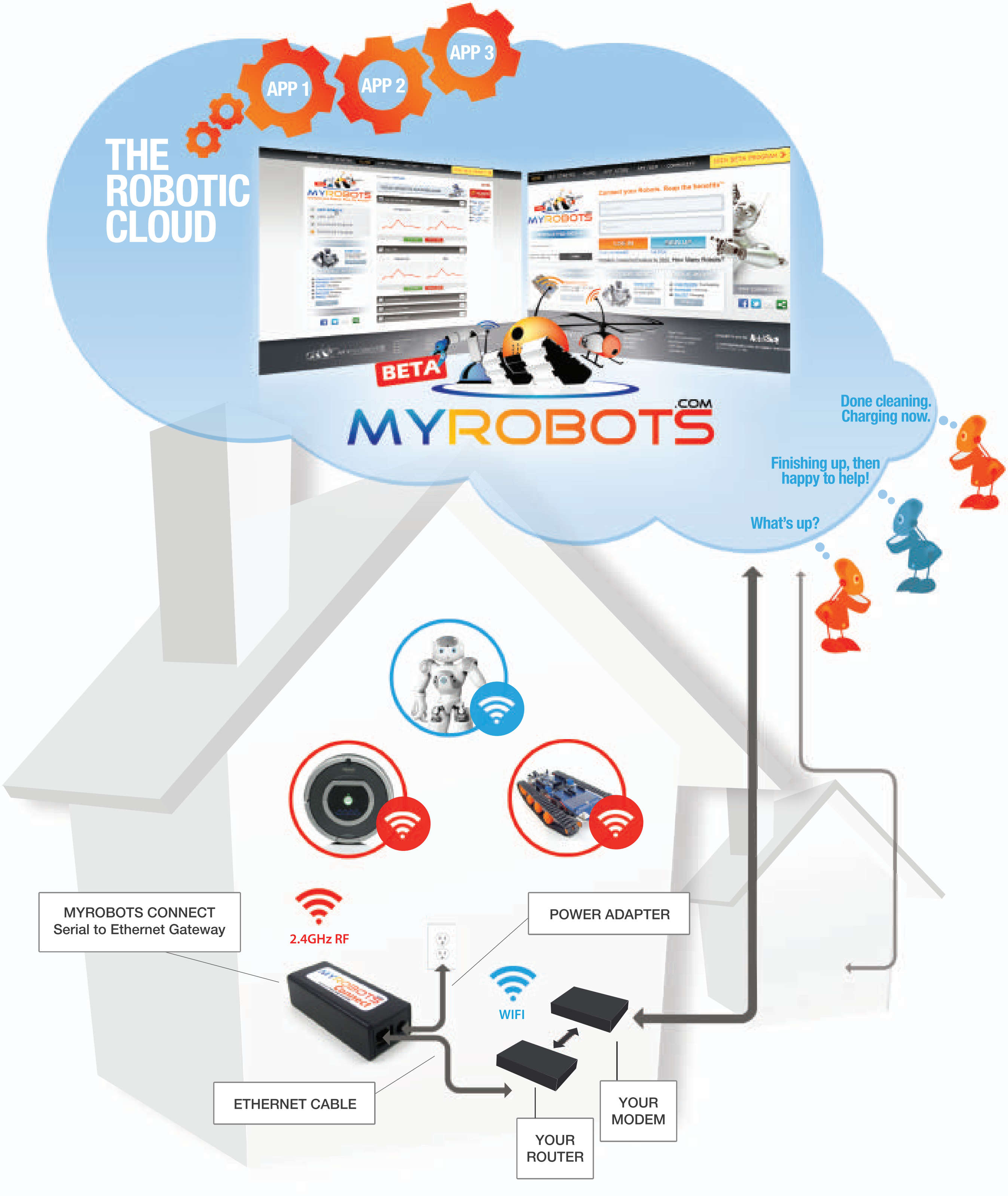
MIDI is a wonderful method of adding music to your robot, but it's not the only technique you can use. A number of Arduino shields and breakout boards allow you to reproduce sound from MP3 and other kinds of digitized clips. Next time around, we'll talk about adding recorded voice, music, and sound effects to your Arduino-based robot. **SV**

Sources

Budget Robotics
ArdBot Chassis: Precut Body Parts, Assembly Hardware
www.budgetrobotics.com

Parallax
Line Follower Module (#28034)
www.parallax.com

SparkFun Electronics
Music Instrument Shield (DEV-10587)
www.sparkfun.com



Connect your robots. Reap the benefits ™

MyRobots.com strives to make cloud robotics a reality accessible to everyone and everything by enabling all robots and smart devices to connect to the Internet.

You can think of MyRobots.com as a social network for robots and smart devices. In the same way humans benefit from socializing, collaborating and sharing, robots can benefit from interacting and sharing their sensors' information, which provides insight on their current state, and allows them to be controlled and monitored remotely.

Through the **MyRobots App Store**, robots augment their capabilities so they can transcend their limitations, allowing them to do more than what they were originally designed for. This means robot owners can reap the full benefits of their robots.

Brought to you by RobotShop.com

Powered by ThingSpeak

The *SERVO* Webstore

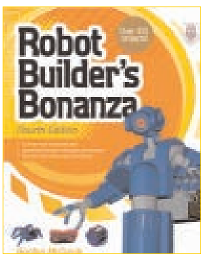
Attention Subscribers ask about your discount on prices marked with an *

CD-ROM SPECIALS

NEW RELEASE

ROBOTICS

Robot Builder's Bonanza, Fourth Edition
by Gordon McComb



Robot Builder's Bonanza, Fourth Edition includes step-by-step plans for a number of motorized platforms. The book is described as a compendium of robotics topics, containing more than 100 projects, including 10 robot designs new to the fourth edition. These modular robots are low cost, and are made to be reproduced by readers with no training in mechanical construction.
$29.95*

Mechanisms and Mechanical Devices Sourcebook 5th Edition
NEW!

by Neil Sclater

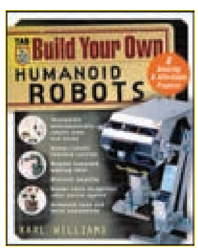
Fully revised throughout, this abundantly illustrated reference describes proven mechanisms and mechanical devices. Each illustration represents a design concept that can easily be recycled for use in new or modified mechanical, electromechanical, or mechatronic products. Tutorials on the basics of mechanisms and motion control systems introduce you to those subjects or act as a refresher.
Reg $89.95 Sale Price $79.95

Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists
by Dustyn Roberts

In *Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists*, you'll learn how to successfully build moving mechanisms through non-technical explanations, examples, and do-it-yourself projects — from kinetic art installations to creative toys to energy-harvesting devices. Photographs, illustrations, screenshots, and images of 3D models are included for each project.
$29.95*

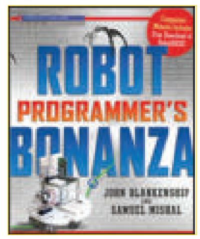
Build Your Own Humanoid Robots
by Karl Williams

GREAT 'DROIDS, INDEED!



This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than $300. Together, they form the essential ingredients for making your own humanoid robot.
$24.95*

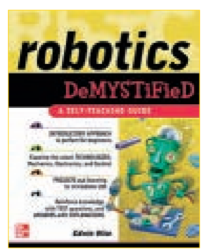
Robot Programmer's Bonanza
by John Blankenship, Samuel Mishal



The first hands-on programming guide for today's robot hobbyist!
Get ready to reach into your programming toolbox and control a robot like never before! *Robot Programmer's Bonanza* is the one-stop guide for everyone from robot novices to advanced hobbyists who are ready to go beyond just building robots and start programming them to perform useful tasks.
$29.95

Robotics Demystified
by Edwin Wise

YOU DON'T NEED ARTIFICIAL INTELLIGENCE TO LEARN ROBOTICS!

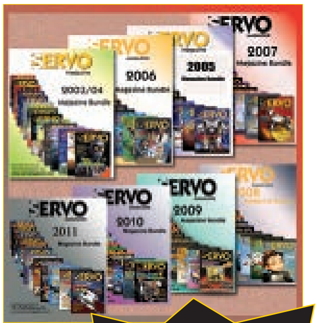


Now anyone with an interest in robotics can gain a deeper understanding — without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots! **$19.95**

We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

To order call 1-800-783-4624

SERVO Magazine Bundles



Save $10.00
Only $57.95!

Now you can get one year's worth of all your favorite articles from *SERVO Magazine* in a convenient bundle of print copies. Available for years 04, 05, 06, 07, 08, 09, 10 and 2011.

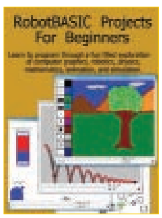
Any bot builders out there?
Get cool robotics stuff from my store!

Call me at my order desk!

Visit my online store @ www.servomagazine.com

RobotBASIC Projects For Beginners

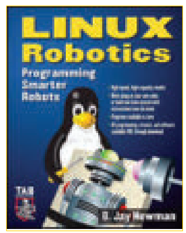
by John Blankenship, Samuel Mishal



If you want to learn how to program, this is the book for you. Most texts on programming offer dry, boring examples that are difficult to follow. In this book, a wide variety of interesting and relevant subjects are explored using a problem-solving methodology that develops logical thinking skills while making learning fun. RobotBASIC is an easy-to-use computer language available for any Windows-based PC and is used throughout the text.
Price $14.95

Linux Robotics

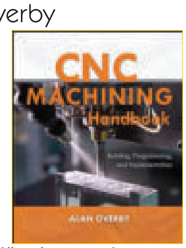
by D. Jay Newman



If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need.
$34.95

CNC Machining Handbook: Building, Programming, and Implementation

by Alan Overby

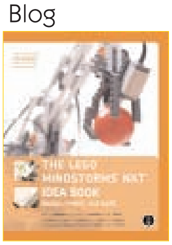


The *CNC Machining Handbook* describes the steps involved in building a CNC machine and successfully implementing it in a real world application. Helpful photos and illustrations are featured throughout. Whether you're a student, hobbyist, or business owner looking to move from a manual manufacturing process to the accuracy and repeatability of what CNC has to offer, you'll benefit from the in-depth information in this comprehensive resource. **$34.95**

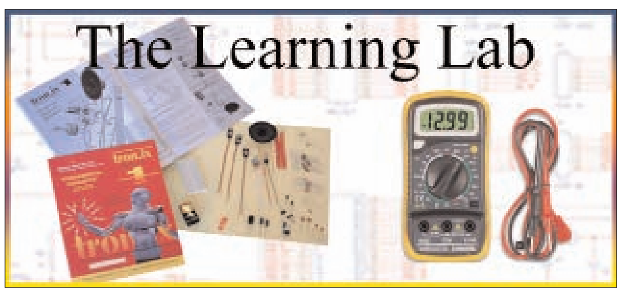
SPECIAL OFFERS

LEGO MINDSTORMS NXT Idea Book

by the Contributors to The NXT Step Blog



If you're serious about having fun with LEGO® robotics, you've come to the right place. The team behind The NXT STEP blog — the authoritative online source for MINDSTORMS® NXT information and advice — has packaged its considerable skills and experience in this book. Inside, you'll find some of the team's best ideas for creating cool and sophisticated models, including instructions for eight robots you can build yourself.
Reg $29.95 Sale Price $24.95



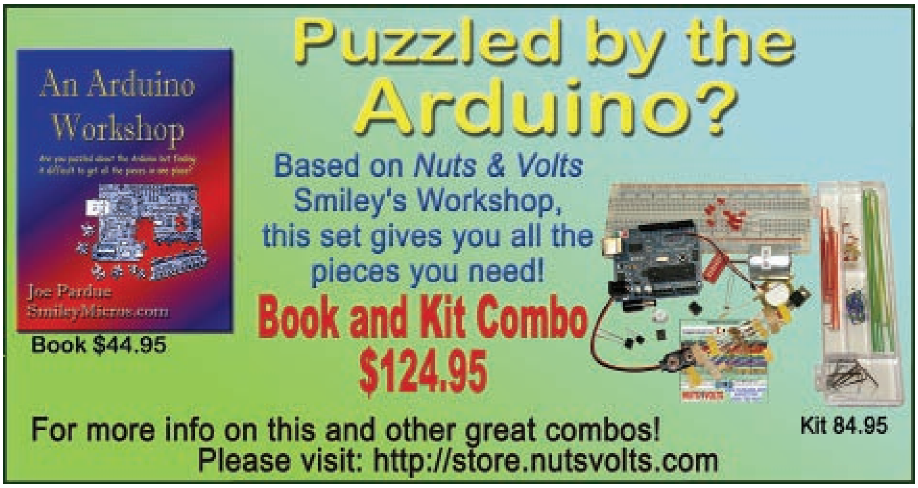
Technology Education Package for Everyone Starting in Electronics

This lab – from the good people at GSS Tech Ed – will show you 40 of the most simple and interesting experiments and lessons you have ever seen on a solderless circuit board. As you do each experiment, you learn how basic components work in a circuit. Along with the purchase of the lab, you will receive a special password to access the fantastic online interactive software to help you fully understand all the electronic principles. For a complete product description and sample software, please visit our webstore.

Regular Price $99.95 **Subscriber's Price $95.95**

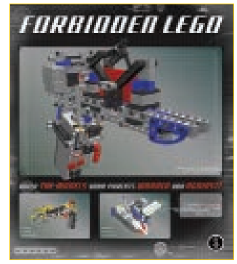
Or order online www.servomagazine.com

SPECIAL OFFERS



Forbidden LEGO

by Ulrik Pilegaard / Mike Dooley

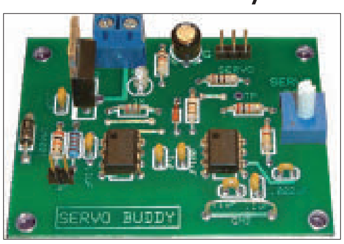


Forbidden LEGO introduces you to the type of free-style building that LEGO's master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse.

Reg $24.95 Sale Price $19.95

PROJECTS

The SERVO Buddy Kit



An inexpensive circuit you can build to control a servo without a microcontroller.

For more information, please check out the May 2008 issue or go to the SERVO webstore.

Includes an article reprint.

Subscriber's Price **$39.55**

Non-Subscriber's Price **$43.95**

3D LED Cube Kit

From the article "Build the 3D LED Matrix Cube" as seen in the August 2011 issue of *Nuts & Volts Magazine*.

This kit shows you how to build a really cool 3D cube with a 4 x 4 x 4 monochromatic LED matrix which has a total of 64 LEDs. The preprogrammed microcontroller that includes 29 patterns that will automatically play with a runtime of approximately 6-1/2 minutes.

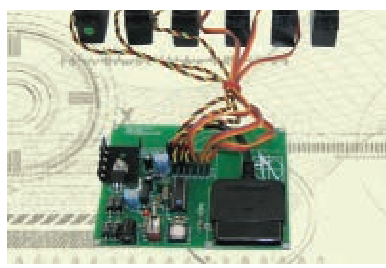
Colors available: Green, Red, Yellow & Blue.

Jig and plastic cases also available.

Subscriber's Price **$57.95**

Non-Subscriber's Price **$59.95**

PS2 Servomotor Controller Kit



This kit accompanied with your own PlayStation controller will allow you to control up to six servomotors. Includes all components and instruction manual.

For more information, please see the February 2011 edition of *SERVO Magazine*. Assembled units available!

Subscriber's Price **$79.95**

Non-Subscriber's Price **$84.95**

Get Your Amateur Radio License with Study Materials from the W5YI Group & Gordon West, WB6NOA

AMATEUR RADIO HAM IT UP! A NATIONAL RESOURCE

TECHNICIAN CLASS

Technician Class book

for the 2010-2014 entry level exam! Gordo reorganizes the Q&A into logical topic groups for easy learning! Key words are highlighted in his explanations to help you understand the material for test success. Web addresses for more than 125 helpful, educational sites. Includes ***On The Air*** CD demonstrating Tech privileges. **GWTM $20.95**

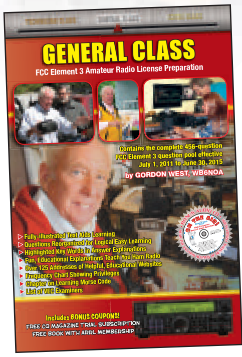
Tech Book & Software Package

Gordo's book with W5YI software allows you to study at your computer and take practice exams. Explanations from Gordo's book are on the software – answer a question wrong and his explanation appears to reinforce your learning. Includes free Part 97 Rule Book. **NCS $29.95**

Tech Audio Course on CD

Welcome to Gordo's classroom! Technician audio theory course recorded by Gordo talks you through the Element 2 question pool. Follows the order of his ***Technician Class*** book, and is full of the sounds of ham radio operating excitement! An excellent study aid if you spend a lot of time in your car or pick-up! On 4 audio CDs. **GWTW $29.95**

GENERAL CLASS



General Class book

Upgrade to the HF bands with Gordo & W5YI! Gordo's manual for 2011-2015 reorganizes all the questions into logical topic groups for easier learning. His explanations include highlighted key words to help you remember the material for test success. Companion CD is full of great operating tips! Available about May 1st. **GWGM $24.95**

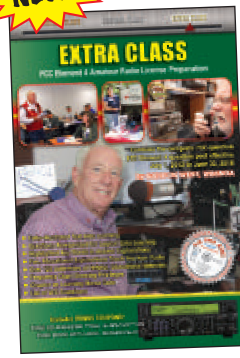
General Book & Software Package

Study at your computer and take practice exams. Software includes explanations from Gordo's book, scores your results and highlights areas that need further study. Includes free Part 97 Rule Book. **GUS $34.95**

General Audio Course on CD

General Theory Course recorded by Gordo is full of the sounds that bring ham radio to life! He talks you through the Element 3 theory to help you understand the material and get you ready for your upcoming exam. On 4 audio CDs. **GWGW $29.95**

NEW! EXTRA CLASS



Extra Class book

Go to the top with Gordo! 2012-2016 book includes all Element 4 Q&A reorganized into logical topic groups. Gordo's fun, educational explanations with highlighted keywords, and great memory tricks for those tough theory questions! Wait 'til you meet "Eli the Ice Man!" **GWEM $24.95**

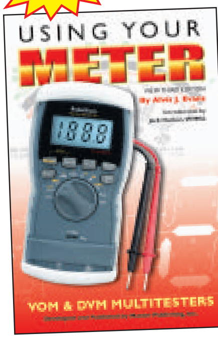
Extra Book & Software Package

Study at your computer and take practice exams as the W5YI software scores your results and highlights areas that need further study. Includes explanations from Gordo's book. Free Part 97 Rule Book. **ECS $39.95**

Extra Audio Course on CD

Extra Class Theory Course recorded by Gordo talks you through the difficult Element 4 theory to help you understand the material and get you ready for your upgrade to the top. On 6 audio CDs. **GWEW $39.95**

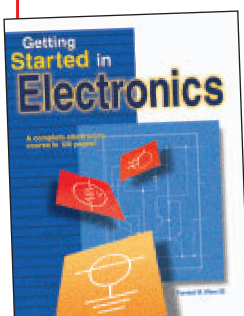
NEW! Using Your Meter



Teach yourself the correct use of your multitester. Book explains fundamental concepts of electricity including conventional and electron current and series and parallel circuits. It teaches how analog and digital meters work and tells you what the voltage, current and, resistance measurements mean. Then it provides fully-illustrated, step-by-step instruction on using your meter in practical applications in the home, workshop, automotive and other settings. An excellent learning tool and reference for the hobbyist and ham. **METR $24.95**

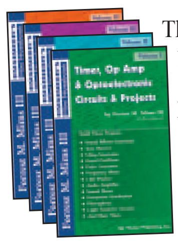
Let Forrest Mimms Teach You Electronics!

Getting Started in Electronics



This is a complete electronics course in 128 pages! This famous electronics inventor teaches you the basics, takes you on a tour of analog and digital components, explains how they work, and shows how they are combined for various applications. Includes circuit assembly tips and 100 electronic circuits you can build and test. Forrest has written dozens of books, hundreds of articles, invented scientific devices and travelled to the Amazon for NASA, and loves to share his knowledge with eager students! **GSTD $19.95**

Engineer's Mini Notebooks



These Mims classics teach you hands-on electronics! Study and build 100s of practical circuits and fun projects. Each volume contains several of his famous Mini Notebooks. Terrific ideas for science fair projects and a great way to learn about electronics! Useful reference guides for your workbench!

Vol. 1: *Timer, Op Amp, & Optoelectronic Circuits & Projects* **MINI-1 $12.95**

Vol. 2: *Science & Communications Circuits & Projects* **MINI-2 $12.95**

Vol. 3: *Electronic Sensor Circuits & Projects* **MINI-3 $12.95**

Vol. 4: *Electronic Formulas, Symbols & Circuits* **MINI-4 $12.95**

Order today from W5YI: 800-669-9594 or on-line: www.w5yi.org

The W5YI Group P.O. Box 200065 Arlington, TX 76006-0065 *Mention ad code NVSM for a free gift.*

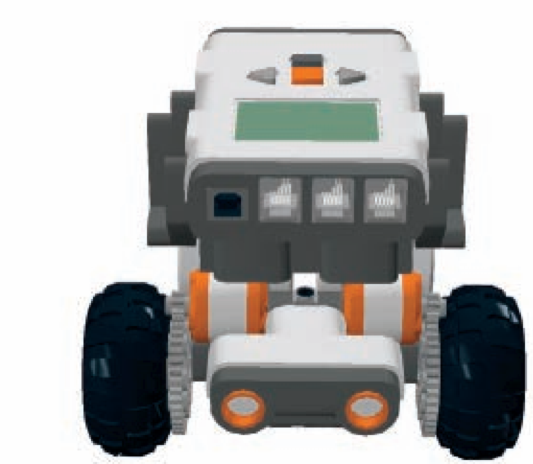
NVSM

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

The NXT Big Thing #18

Chariot of Fire

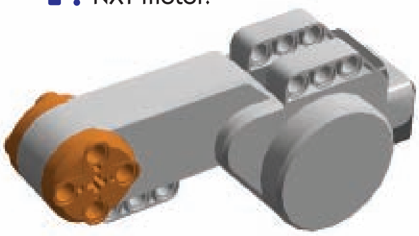
By Greg Intermaggio

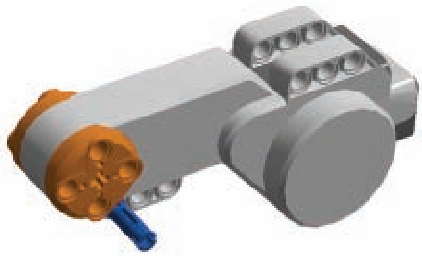


Hullabaloo and howdy doo, and welcome to the 18th installment of The NXT Big Thing! This month, we'll dive right into creating an awesome Sumo bot called Chariot of Fire which you can use to compete. You can also use Chariot of Fire as inspiration for your own LEGO Sumo robot! With RoboGames just around the corner (**robogames.net**), it's time to hunker down and get your bots ready for battle!

Building Instructions: Chariot of Fire

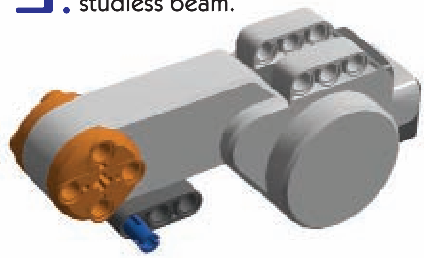
1. Start with just an NXT motor.





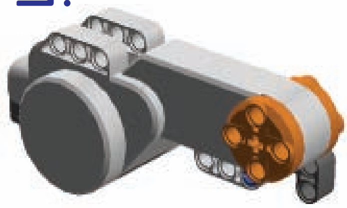
2. Add a double friction pin (these come in both blue and black).

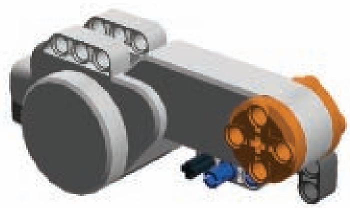
3. Slide in a three-hole studless beam.



4. Attach a 4 x 2 angled studless beam.

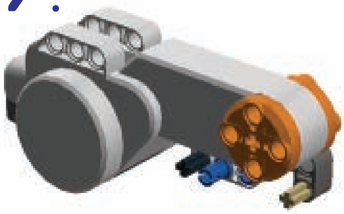
5. Re-orient the assembly.



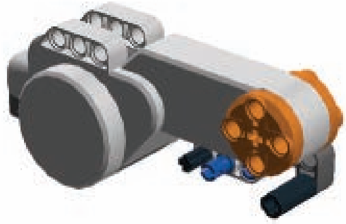


6. Slide in a four-stud axle and snap in a double friction pin as shown.

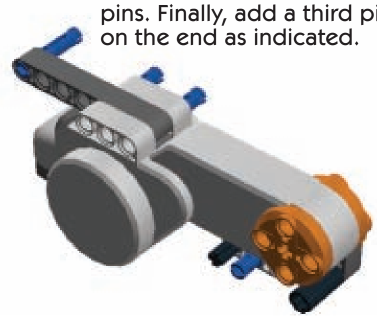
7. Snap in a hybrid axle-pin.

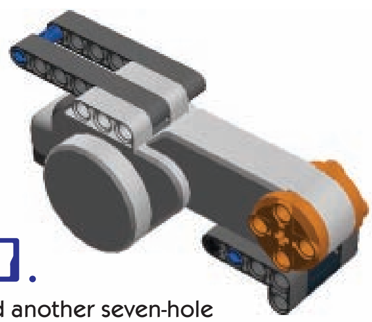


8. Add an axle extender.

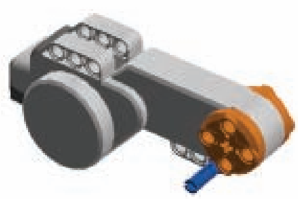


9. Slide in a seven-hole studless beam, then secure it with two double friction pins. Finally, add a third pin on the end as indicated.

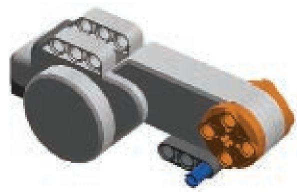




10. Add another seven-hole studless beam and another 4 x 2 angled studless beam.



11. Build the same motor assembly in Step 2 in reverse.



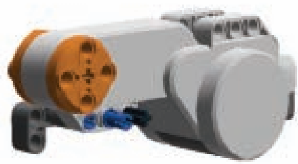
12. Build the same motor assembly in Step 3 in reverse.

13. Build the same motor assembly in Step 4 in reverse.

14.

Build the same motor assembly in Step 5 in reverse.

15. Build the same motor assembly in Step 6 in reverse.

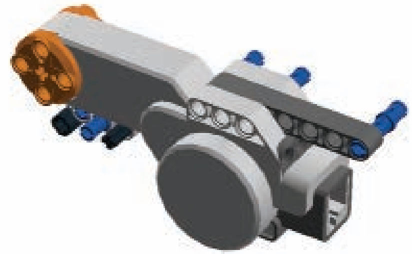


16.

Build the same motor assembly in Step 7 in reverse.

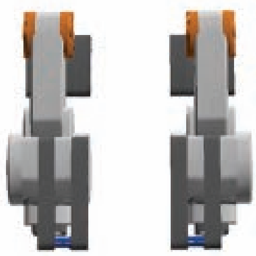
17. Build the same motor assembly in Step 8 in reverse.

18. Build the same motor assembly in Step 9 in reverse.



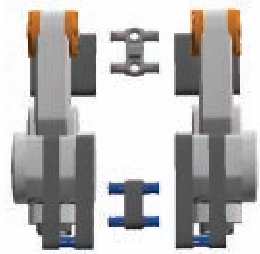
19. Build the same motor assembly in Step 10 in reverse.

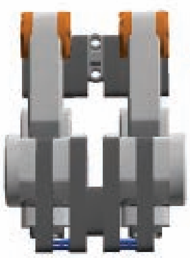
20.



Align the two motor assemblies next to each other.

21. Prepare a combination of the double pin shown, and/or your own similar assembly with two double friction pins and a three-hole studless beam.





22.

Use those pieces to secure the two motor assemblies.

23.

Re-orient the assembly.

24.

Slide a four-stud axle through each motor and a five-stud axle through either axle extender. Then, attach eight-spur gears to the motors and 40-spur gears to the axle extenders. This will be the drivetrain of our robot.

25.

Snap in two friction pins on each side of the assembly, and grab an NXT (note: This model is designed for use with AAs- not the rechargeable battery). Snap a blue hybrid axle-pin to either side of the NXT.

26.

Pop in a double angled studless beam and a five-hole studless beam on either side of the NXT as shown.

27.

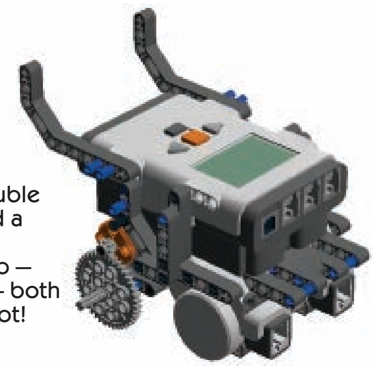
Snap in three standard friction pins and one double friction pin on each side of the robot.

28.

Add a 5 x 3 angled studless beam, a friction pin, and a double friction pin on both sides of the robot.

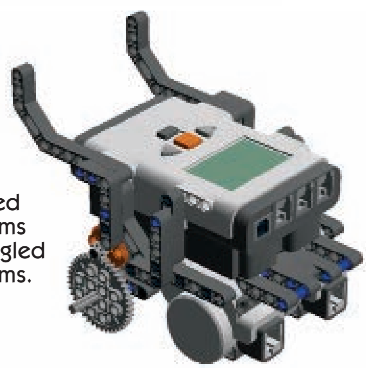
29.

Snap in two double friction pins and a double angled studless beam to — you guessed it — both sides of the robot!



30.

Secure the double angled studless beams with 5 x 3 angled studless beams.

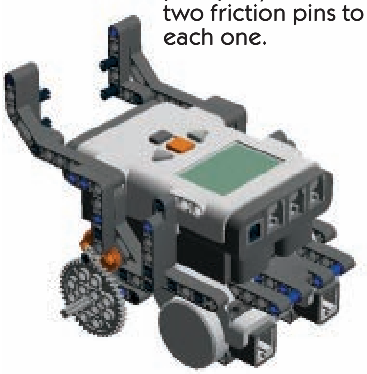


31.

Slide blue hybrid axle pins into the double angled studless beams, and snap in friction pins as shown.

32.

Attach 5 x 3 angled studless beams to your pins, then add two friction pins to each one.

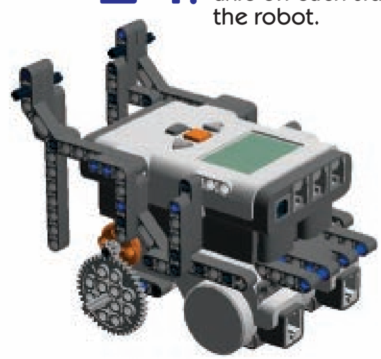


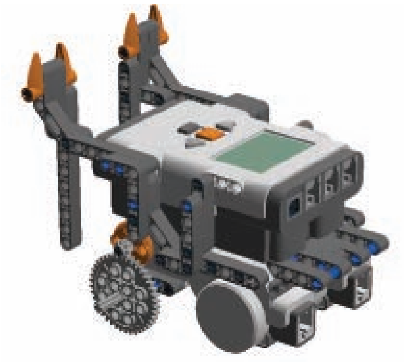
33.

Snap 11-hole studless beams into your pins on either side. These will allow Chariot to run your opponents off of a raise ring without falling out itself!

34.

Slide in a four-stud axle on each side of the robot.



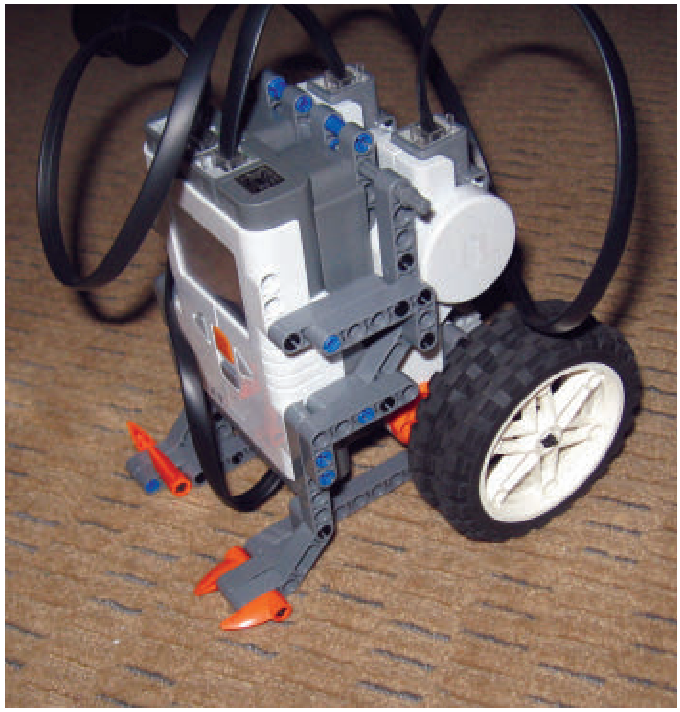


35.

Attach four orange Bionical eyes as shown. These will be used to get under your opponents!

36.

Re-orient the robot, and attach a light sensor using two friction pins.



37.

Finally, attach two motorcycle tires to each side of the robot. Plug the left motor into port B, the right into port C, and the light sensor into sensor port 3, and you're done!

Awesome! Now all it needs is a good program! Check out the previous NXT Big Thing #3 or #7 columns for instructions on basic Sumo programming, or come up with your own more complex solution! See you next time! **SV**

Greg "LEGO" Intermaggio lives in the Bay Area, CA, where he runs a business called Techsplosion, bringing hands-on science to all ages and walks of life. In his spare time, Greg likes to unicycle, juggle, unicycle juggle, and battle killer robots! More information about Greg can be found at **Intermaggio.com**. More information about Techsplosion can be found at **Techsplosion.org**.

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

Then and NOW

Robot-Assisted Prostate Surgery

by Tom Carroll

Cancer! That's a word we all dread to hear, whether about someone we know or especially concerning ourselves. I heard that word spoken to me early last year after I had a prostate biopsy. Needless to say, thousands of things were spinning through my mind. I knew that prostate cancer is the second most common cancer among men worldwide, with one in six men here in the US diagnosed with the disease. Another scary fact is that prostate cancer is the second leading cause of cancer deaths among US men. The good news is if the cancer is discovered early, there is a much greater chance of curing prostate cancer. "What is the next step?" I asked my doctor, as I was aware that there were many options available to men with this condition.

FIGURE 1. Intuitive Surgical da Vinci robot.

FIGURE 2. Michael Lipke, MD at the console of the da Vinci system.

As a reader of a robotics magazine, I'd like to take you on a trip into an operating room with a robot performing surgery on a person: me. Before discussing robot-assisted prostate surgery, I'd like to go back a bit to my first prostate biopsy back in 2002. Understanding prostate health is important in understanding the need for any sort of treatment option. I had a PSA (prostate specific antigen) blood test score of 4.7 and it had been slowly rising from 3.2 to 3.4 in the late 1990s. My family practice doctor was concerned and sent me to a urologist. With data in hand, my urologist performed the needle biopsies and he found no cancer. Yes, the days waiting after the biopsy until he told me the 'good' results were not days that I enjoyed. Nine years later and a new doctor, my PSA had risen to 11.95 and another needle biopsy was necessary.

There has been much speculation about the need for a PSA test recently, and even some doctors raised concern that the test could actually be harmful since a few patients have incurred severe infections from the biopsy procedure. Some patients have been rushed into surgery only to find tiny scattered cells that would not have grown into tumors for years. I happen to be in the majority who believe in these tests and the subsequent biopsies. The needle biopsies are not enjoyable but can discover a cancer situation where subsequent surgical

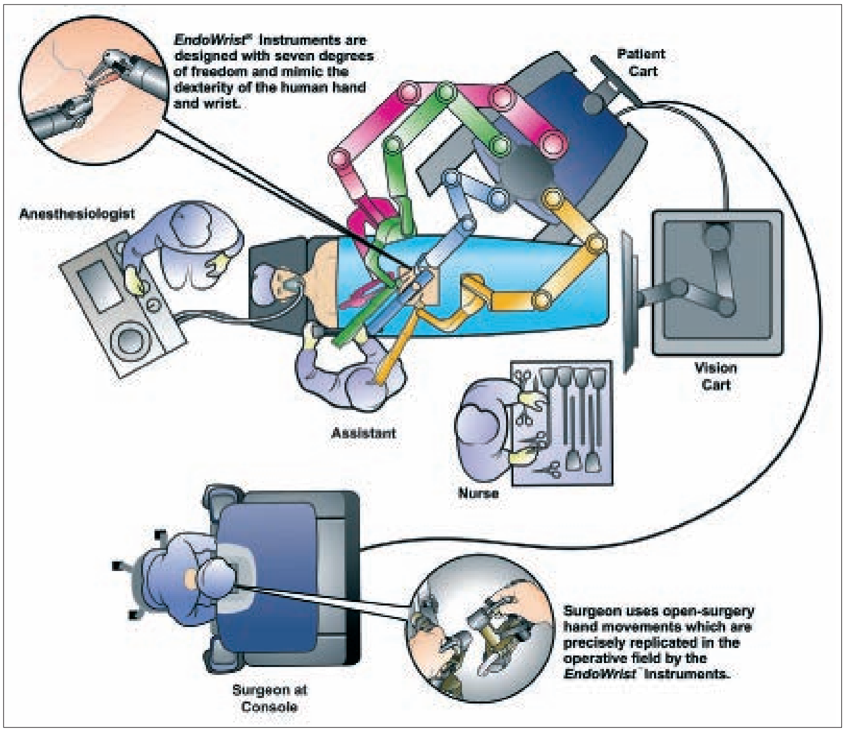


FIGURE 3. da Vinci surgical system operating room setup.

procedures can be performed to save a life — *my* life.

The Options

There are many options that urologists have at their disposal, though most of these specialists prefer to use only one or two of them in the treatment of their patients. One, of course, is to wait and see. Prostate cancer is usually a very slow growing type of cancer and many times a man may die of some other age-related disease before the prostate cancer could ever be fatal.

Standard surgical removal of the prostate and seminal vesicles via the tried and true open surgery method is also an option, though the hospital recovery period is longer. Doctors can also use laparoscopic methods wherein smaller incisions are used to enter the body at the site of the prostate, and handheld small instruments are then inserted, along with a light and vision system (fiber optic or television) to visualize, manipulate, and cut and remove tissue. Robotic surgery is similar to laparoscopic, but better control and magnification is possible.

Radiation therapy — either implanted radioactive seeds or external beam — to kill the cancerous tissue is another method. Cryotherapy is used since it does kill the complete prostate gland and any cancer cells, but it has some severe side effects. Hormone therapy and chemotherapy are used in many cases; a good friend of mine swears by his treatments.

Your doctor will discuss these options with you and may strongly suggest the method he/she prefers. My doctor chose to completely remove my cancerous prostate and the nearby seminal vesicles as my treatment option. If you find yourself facing any sort of cancer possibility and feel uncomfortable with the methods your doctor discusses, of course, get a second opinion.

Robot-Assisted Prostate Surgery

Six years ago, I wrote an article for this column about robotic surgery, having suspected a possibility of having prostate cancer sometime in the future after my first biopsy in 2002. I began looking at various options and quickly settled on a robot-assisted prostatectomy as the best choice for me. The choice was not made because of my robotics background, but due to the less invasive method employed. I found out that my present urologist (Dr. Michael Lipke) had previously trained on the Intuitive Surgical da Vinci system shown in **Figure 1** and had performed hundreds of successful prostate surgeries. Despite the terminology, the da Vinci 'robot' is not a true intelligent or autonomous robot, but is a *tele*-robot controlled by a surgeon from a nearby console (see **Figure 2**); as the title implies, it's robot *assisted* surgery. **Figure 3** shows the general operating room setup. My surgeon used the da Vinci "S" HD system, though the hospital also had a newer da Vinci "Si" surgical system.

Training to use the da Vinci system is getting easier. Mimic Technologies is developing MScore — a portable trainer device shown in **Figure 4**. As stated from MScore: "The MScore system allows every movement and action the <prospective> surgeon makes to be tracked and evaluated within a virtual reality training environment. A surgeon's proficiency and score is established by utilizing a wide variety of performance metrics, such as task time, efficiency of instrument motion, blood loss, and the force applied to

FIGURE 4. Mimic Technologies da Vinci trainer.

FIGURE 5. Ankle arthroscopy instruments.

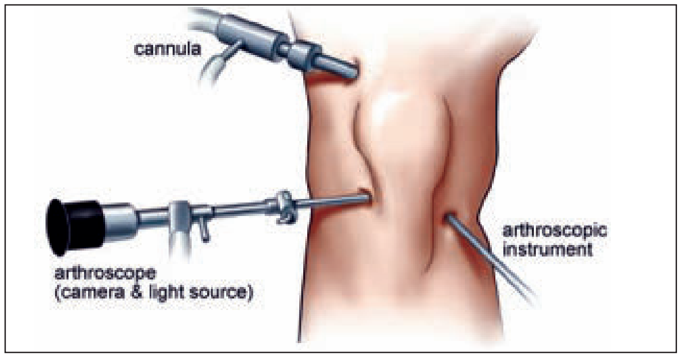


FIGURE 6. Arthroscopic knee surgery.

tissue. Performance baselines are derived from the data collected from experienced surgeons." This type of trainer allows surgeons to practice and improve without taking up valuable and expensive time on the actual machine.

How are Various Surgeries Performed?

To understand robot-assisted surgery, you must sit back and imagine how surgeons perform any type of surgery on a human. In typical open procedures, the area to be operated on is fairly wide open and exposed, with clamps to hold tissue in place and to make it easier to apply suction to remove fluid buildup. A large overhead lamp illuminates the area. The surgeon can then use scalpels, clamps, forceps, hemostats, and other instruments that can easily be manually manipulated in whatever way necessary. The surgeon is usually standing up and looking down on the surgery site.

Now, imagine a more localized surgery such as a shoulder or knee joint where the surgeon elects to use arthroscopic techniques to operate through small holes in the skin and tissue. Arthro meaning 'joint' and scopic meaning 'vision' surgery is a technique whereby the doctor uses handheld instruments to operate within the body without performing major open surgery; a few of which are shown in **Figure 5**. **Figure 6** shows knee surgery using a cannula to circulate saline solution within the operating area, an arthroscope to allow the surgeon to see within the joint area, and the particular surgical instrument. Sometimes the term orthoscopic surgery is used, but this is a broad term implying clear vision surgery. As you can imagine, the surgeon can maneuver the instrument around enough to get to the small areas required for surgery. The outer skin and tissue can move a bit to allow entry from slightly different directions. This type of surgery is also called laparoscopic surgery (or minimally invasive surgery; MIS) when it is performed in other areas of the body.

The Case for Robot-Assisted Surgery

Now, envision an operating site deep with the body — a location such as

FIGURE 7. Being wheeled into the operating room on a gurney.

the prostate and bladder — that could only be reached by open surgery techniques. The depth of the surgery prevents ease of movement with the arthroscopic instruments without making large skin/tissue incisions. The surgeon now requires many axes of instrument movement — not available with handheld arthroscopic instruments. If a particular surgery requires 90 degree positioning of scissors or forceps to reach a particular area of interest, the surgeon could enter from another hole in the body, or use an instrument that can rotate at the end with several degrees of freedom (such as many industrial and experimental robot arms). This is the technique practiced by surgeons using the da Vinci system.

With the operating site so deep within the patient's body and only small holes in the skin and underlying tissue available, the surgeon is limited in how much he or she can actually move the ends of the EndoWrist instruments — the working 'hands' of the robotic system (more on this in a moment).

Many months of intensive training is required before a surgeon begins to feel at ease with using a console 10 feet away from the patient to perform delicate surgeries. Instead of observing a bladder that is the size of a baseball to a softball with a plum-sized prostate gland attached to it several feet below his face on an operating table, the surgeon can now see in high-definition 3D a bladder that is the size of a soccer ball with a prostate gland the size of a baseball directly in front of his face.

FIGURE 8. 'Out cold' on the table, draped in cloths, and ready for the robot.

Pre-Operation Procedures

After my initial check-in at 5:30 in the morning, I was given a gown, prepped in the pre-op room, hooked up to several IVs, and wheeled into the operating room on a gurney. Quite frankly, I don't remember anything after the pre-op. I could have been signing over my first-born and my house for all I knew (**Figure 7**) as everyone in the operating room was crowding around me.

Before the actual procedure, I was transferred to an operating table and my body was tipped head down to allow my gastrointestinal system to move away from the area of the operation. The anesthesiologist had to adjust the respiratory pressure because of this position to keep my lungs at full capacity. **Figure 8** shows how I was draped in cloth and clear plastic in preparation for the surgery.

How the EndoWrist Surgical Instruments Work

Before talking about the actual surgery, let me describe the instruments and how they work when attached to the robot. **Figure 9** shows three of the EndoWrist instruments given to me by my surgeon. At $2K to $2.5K each, these instruments can only be used 10 times. This limit was set

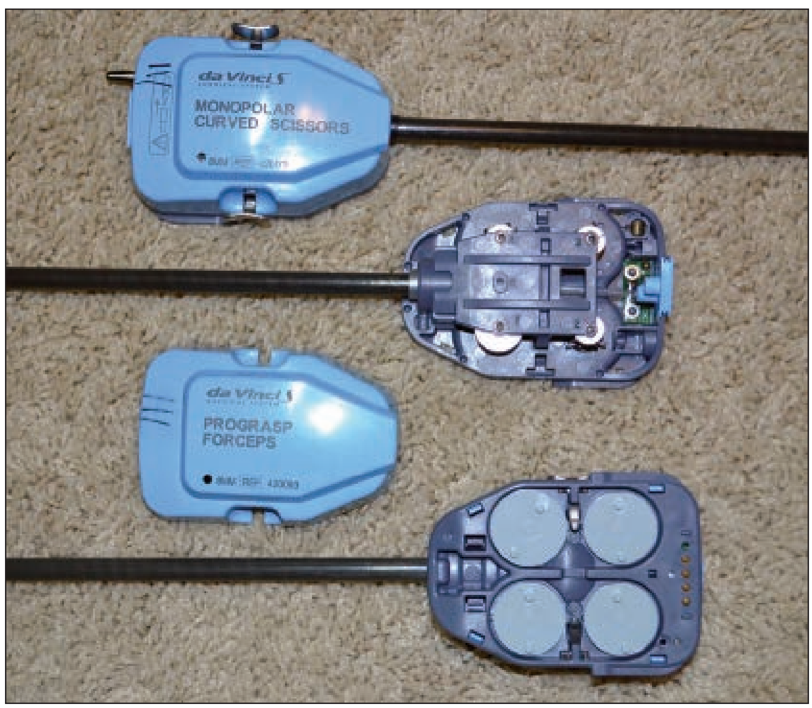


FIGURE 9. Three EndoWrist instruments; one with the cover removed and one with the interconnecting disks shown.

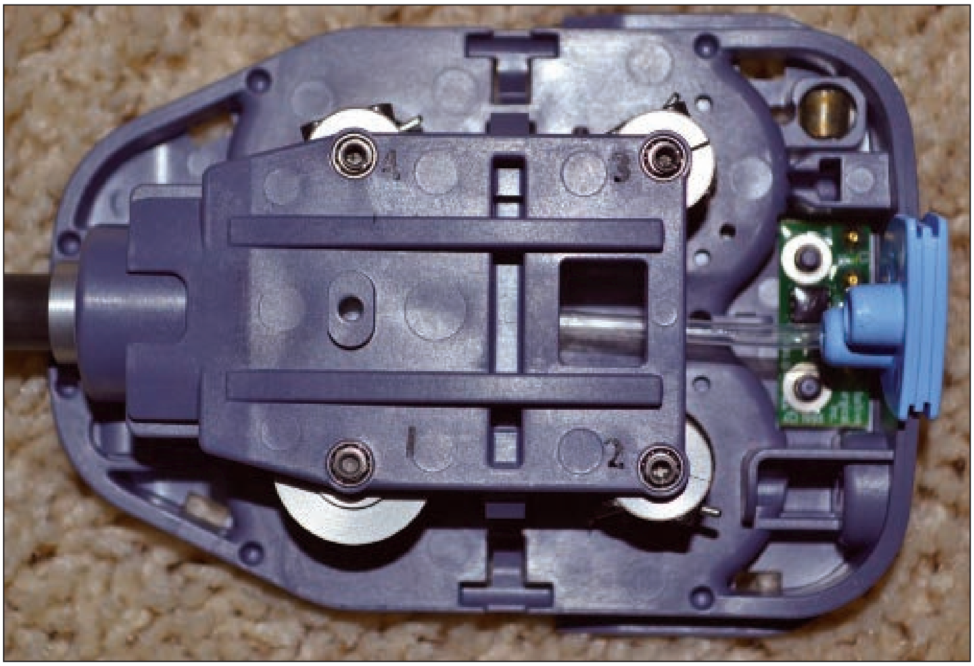


FIGURE 10. EndoWrist head showing cable reels.

FIGURE 11. A typical trocar 'tube' to allow instruments to easily enter the body. The inner pointed part is removed after insertion.

(not because they get worn out since most EndoWrists can last much longer) to insure a very safe limit for a device that is used within a human body. At this point, the tiny memory chip on the green circuit board inside the head of the EndoWrist (shown in **Figure 10**) has recorded each surgical use and it informs the da Vinci system that it can no longer be used (or it can be given to an overly-curious patient like me). The blue plastic port and attached clear tubing above the circuit board allows a saline solution to be circulated at the EndoWrist's working site.

Before I was given the EndoWrist instruments, I assumed that the actual end-effector received its mechanical movement via a series of rotating carbon fiber tubes; the smallest fitting within a slightly larger diameter tube, and so on. I discovered that the EndoWrist uses four small reels that move four stainless steel cables in or out that run up the carbon fiber tube to the end-effector. The largest diameter reel (# 1 in **Figure 10**) moves a short cable that runs to another smaller reel about 5 mm below, whose axis is 90° to the larger and faces down the outer tube; this causes the whole tube and end-effector to rotate almost 520°. The cables from two smaller reels at the far end of the blue attachment box (#2 and #3 in **Figure 10**) are each fastened to one of the two sides of forceps or scissors. Rather than have one reel open and close both halves of a pair of forceps or scissor tips, one reel/cable controls each half of the 'jaws' and allows the forceps or scissors to swivel and cut/pinch in a 200°+ arc. The fourth small reel's (#4 in **Figure 10**) set of cables moves the tip in a 180º arc at a 90° angle to the jaw motion.

So, the surgeon has four degrees of freedom with the four cable reels, and a fifth axis which gives the robot the ability to move the instrument in and out along the axis of the tube. There are actually two more axes of motion — though limited — with the ability to tilt the instrument back and forth in an x-y motion. With a choice of three instruments to manipulate, hold, cut, pinch, and perform other functions, plus a high definition, magnified, and stereoscopic view of the operating site through the EndoScope camera system, many surgeons say that "this is the only way to perform delicate surgeries such as a prostatectomy or hysterectomy." Another plus over handheld instruments using arthroscopic or laparoscopic procedures is that the surgeon does not have to hold his or her various instruments in a certain position for long periods of time — the robot does it. Doctors can rest a moment and determine the next step in the operation.

Positioning of the Robotic EndoWrist End-effectors

You may be wondering how the instruments are positioned into the surgery area. Obviously, a surgeon cannot have the robot blindly insert an instrument into a patient. The instruments must be positioned at the operation site without damaging nearby tissue or organs. Trocars are the guiding tubes that the surgeon inserts through the abdominal wall into the operating area. **Figure 11** shows a typical trocar with the pointed insert. **Figure 12** shows my personal trocar sites; these are the small entry sites for the surgical instruments. Six small incisions were made in my skin. The photo looks a bit scary but the scars have all but disappeared. This is a delicate manual procedure because the surgeon must not puncture any arteries, veins, or organs. The pointed end of the trocar moves slowly through tissue to the operation site; the point is then collapsed and removed so the surgeon can later insert, manipulate, and remove the EndoWrist instrument.

My surgeon used trocars of several different sizes to create entry sites ranging from 5, 8, and 12 mm in diameter. The largest incision (12 mm which was later extended to 4-5 cm) that was made on my body was just above my navel; this was first used for the camera, and then was extended to remove my prostate and seminal vesicles.

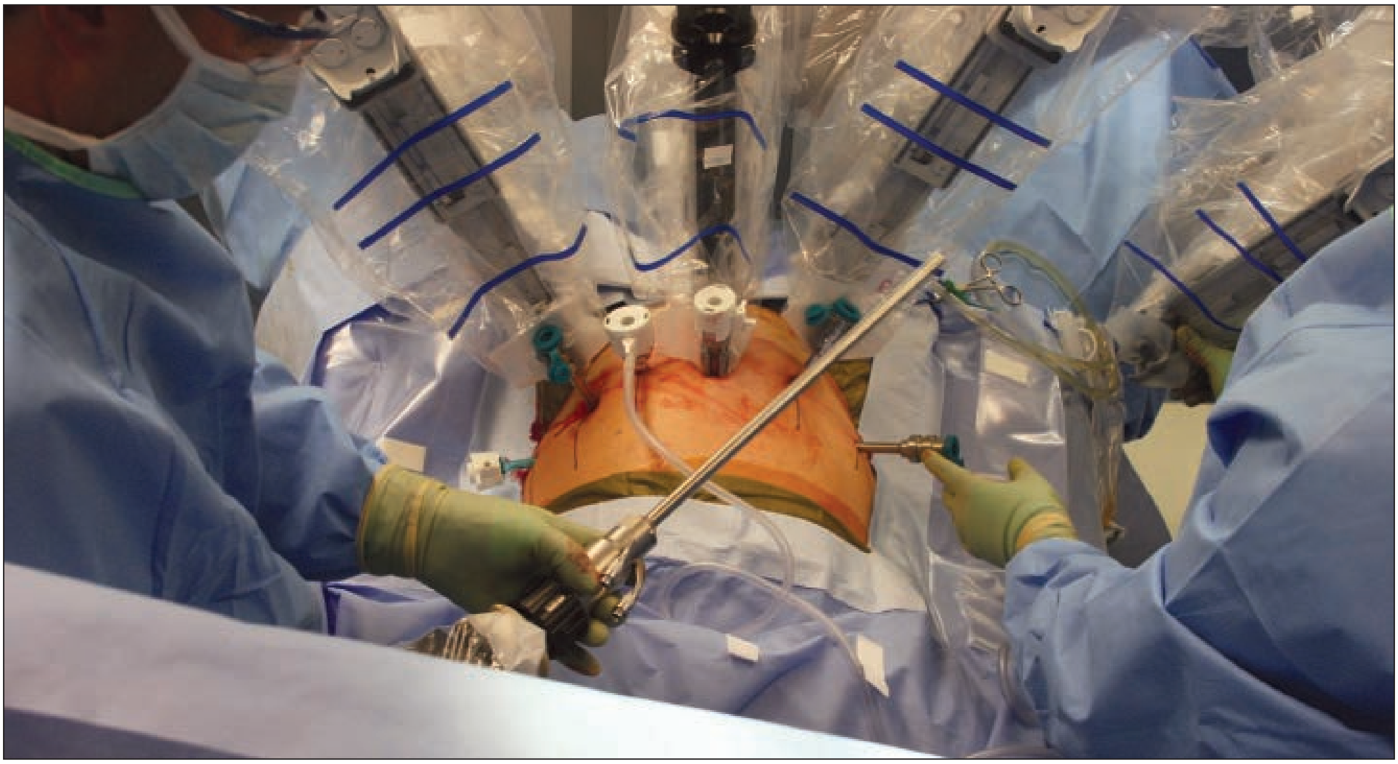


FIGURE 12. The robot is aligned with the trocars.

Smaller trocars were used for the 8 mm EndoWrist instruments, then another was used as a drain. A second 12 mm trocar was used as an assistant port.

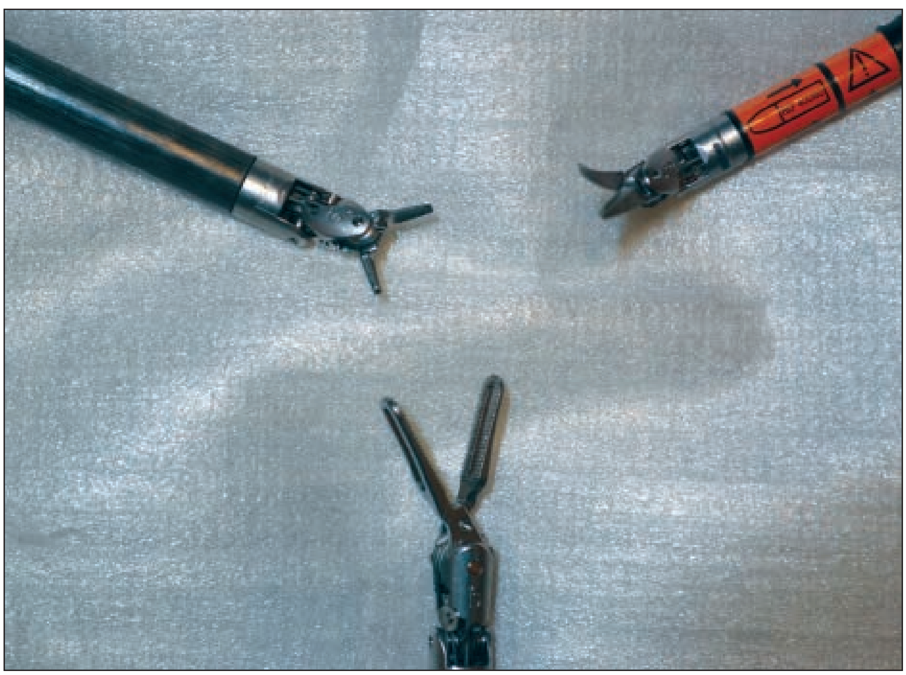
Notice how my abdomen is slightly inflated with CO_2 gas through the attached clear tubing to allow an open view of the prostate site and operating area with the endoscope TV camera. The first trocar in was the 'optical' trocar so my surgeon could see what was happening with the insertion of the remaining trocars and the required operating instruments.

The Surgery Procedures

With the trocars in place, a catheter hooked up, the abdomen inflated at 15 mm/Hg of pressure, and the endoscope video and lighting in place, the da Vinci robot is brought over and locked into position over the patient. Each small step is certainly more complex than the following description, but I'm just giving a quick timeline. The required EndoWrist instruments for the first part of the surgery are slowly inserted through the trocar and into the operation area. The surgeon first severs the vas deferens and the seminal vesicles, and removes them. The bladder is lowered to reach the prostate. The delicate procedure of severing the prostate from the bladder is performed as the surgeon must not damage the bladder's sphincter muscle that acts as a valve for empting the bladder. The urethra is carefully severed from the other side of the prostate and the gland is put into a small 'zip lock bag' to be removed. The urethra is then stitched to the sphincter muscle to get the patient's plumbing back in order.

All of this takes from two and a half to three hours. Just imagine the many hundreds of critical movements that the surgeon has to perform! There are many instruments

FIGURE 13. Three different EndoWrist end-effectors.



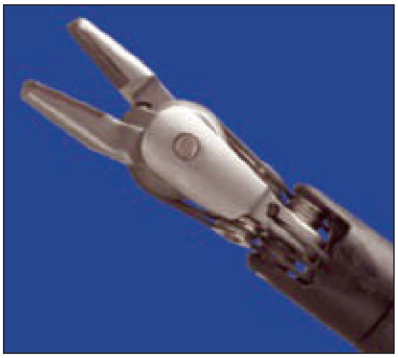


FIGURE 14. EndoWrist needle driver.

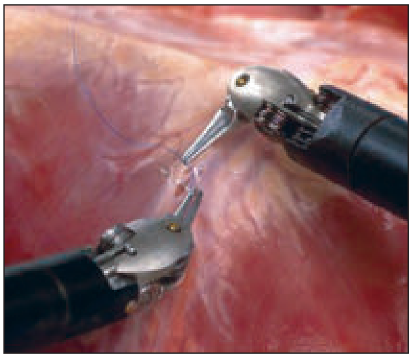


FIGURE 16. A basic suture procedure.

that he/she must use, such as needle drivers, forceps, scissors, and many others. **Figure 13** shows a close-up view of my three EndoWrist instrument's 'working' ends. **Figure 14** shows a close-up view of a needle driver EndoWrist. **Figure 15** shows a small assortment of the different types of EndoWrist instruments available. Notice how the cables are attached to the different end instruments. Each new instrument is attached to the robot's arm and inserted into the trocar which has a seal at the bottom of the tube. **Figure 16** shows a basic suturing procedure under magnification.

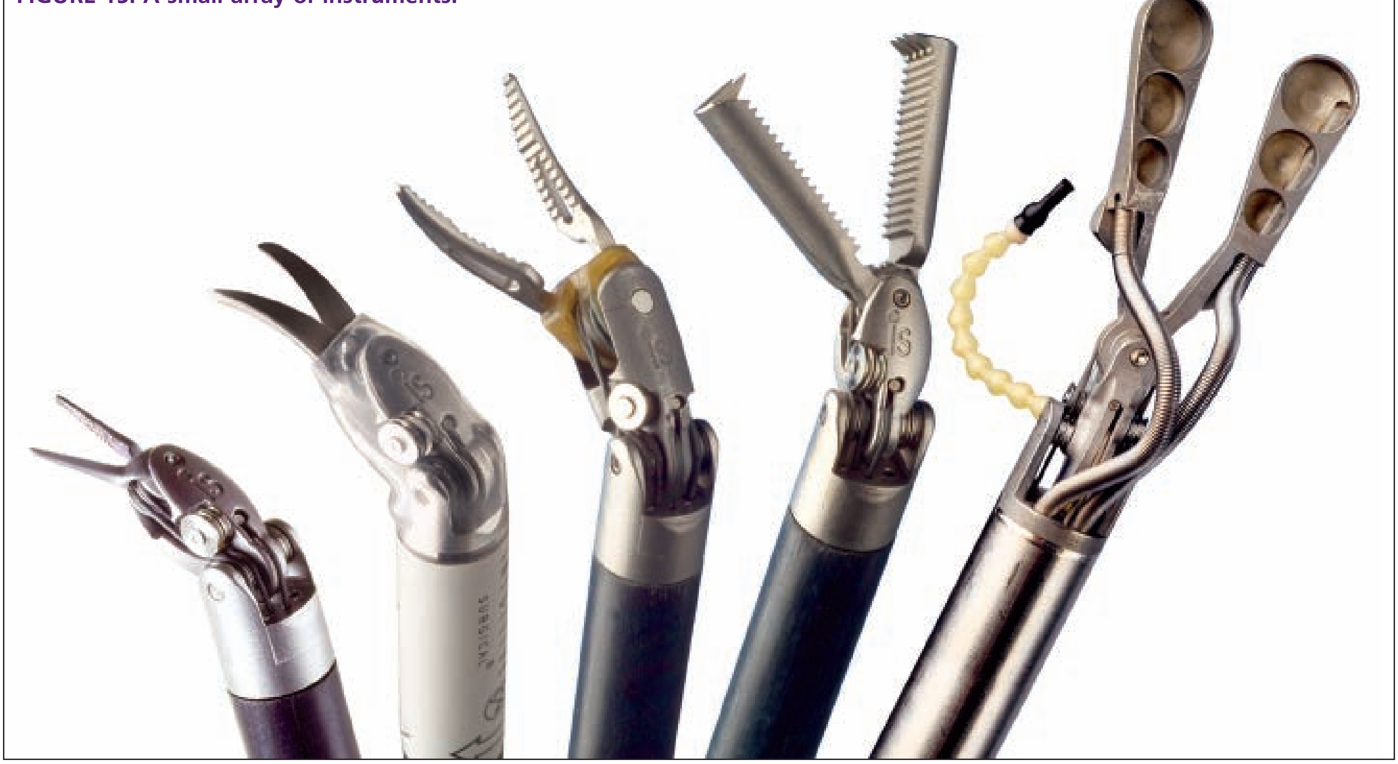
My operation could have been accomplished by any of the methods I talked about earlier. I preferred the da Vinci robot-assisted surgery, mainly for the ability of my surgeon to do the delicate procedures such as nerve sparing and small suturing to allow me a quick recovery process. This minimally invasive procedure allowed me to be up and walking the afternoon of my surgery, and going home the next day. Yes, there were a few more days of rest and simple exercising, but I was soon doing daily hikes with my wife.

Final Thoughts

In a demonstration, engineering and post-grad computer science students at Johns Hopkins University in Baltimore modified a Da Vinci surgical system console and used it as a workstation to control an industrial robot at NASA Goddard Flight Center 30 miles away. The same high definition video link used for surgery allowed operators to see the interior of a satellite, giving them the same virtual sensation of 'being there' that surgeons experience. It is hoped that the repair, refueling, and maintenance of on-orbit satellites may not require the presence of astronauts in the near future. NASA has the Robonaut 2, and possibly it can be used to hand the da Vinci robot tools as needed for space-borne repairs.

I am indebted to my surgeon, Dr. Michael Lipke (a urologist with Kaiser Permanente), who endured my incessant medical and technical questions, and did a great surgery for me, as well as the many nurses, anesthesiologist, assistant surgeon, and other caregivers at Providence St. Vincent Medical Center in Portland. I would also like to thank Providence St. Vincent for allowing more than 60 photographs to be taken during my operation. I appreciate all of the help that Intuitive Surgical gave to me, and steering me to the right people. This was a most unique and educational experience for me. SV

FIGURE 15. A small array of instruments.



ROBO-LINKS

ADVERTISER INDEX

THE USA SCIENCE & ENGINEERING FESTIVAL IS PROUD TO HOST THE 2012 "NATIONAL ROBOT FEST AND DIY EXPO - WHERE CREATIVITY & TECHNOLOGY MEET"

GRAND FINALE

EXPO & BOOK FAIR

APRIL 28 & 29, 2012

WALTER E. WASHINGTON CONVENTION CENTER, WASHINGTON, D.C.

USASCIENCEFESTIVAL.ORG

OVER 3000 FUN HANDS-ON ACTIVITIES AND MORE THAN 100 STAGE SHOWS
MEET AWARD-WINNING AUTHORS AND SCIENCE CELEBRITIES LIKE BILL NYE THE SCIENCE GUY
NEW THIS YEAR: CAREER PAVILION & BOOK FAIR | A FREE EVENT

THANKS TO OUR SPONSORS

LOCKHEEDIUM / FESTIVAL HOST

LOCKHEED MARTIN

BOCKIUM

Larry and Diane Bock

K&L GATESIUM

K&L GATES

NOBELIUM

Northrop Grumman Foundation, Scientific American, Popular Science, NewScientist, EE Times iGEN, technology review (Published by MIT), Popular Mechanics

EINSTEINIUM

C&EN, ResMed Foundation, at&t, Microsoft, life technologies Foundation, National Institutes of Health, Dow, NSF, The Epoch Times, Engineering.com, Farrell Family Foundation

TAKE THE METROBUS OR METRORAIL TO THE USA SCIENCE & ENGINEERING FESTIVAL

KRYPTON

LEGO Education, CrazyEngineers.com, The Kavli Foundation, Medimmune, Sigma Xi, American Scientist, Illumina Inc, Physics Today, Forbes/Wolfe, SchoolTube, National Aeronautics and Space Administration, The Planetary Society, PBS Kids, Washington Family Magazine, George Washington University, WAMU 88.5 - American University Radio

PLATINUM

Vertex Pharmaceuticals, Celestron, Baxter International, American Nuclear Society, Association of Science-Technology Centers, Xconomy, You Can Do the Rubik's Cube, Amgen, U.S. Environmental Protection Agency, U.S. Department of Defense, Society for Maintenance and Reliability Professionals (SMRP), 3M, Northern Virginia Technology Council, Purdue University, Aldebaran Robotics, Inc., Johns Hopkins University Applied Physics Laboratory, Raytheon, Center for America, U.S. Army RDECOM and eCYBERMISSION, Project Lead The Way (PLTW), The Scripps Foundation for Science and the Environment, Center for Biotechnology Education at Johns Hopkins University,

Life with Robot!
Not a dream anymore

THE MASTERPIECE
of ROBOT SERVO

24kgf.cm @ 7.4V
[333.6 ozf.in.]

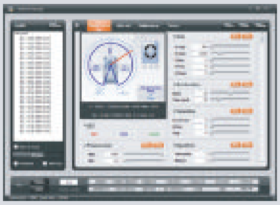
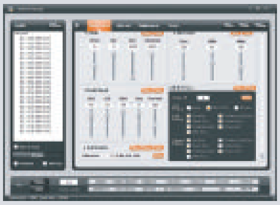
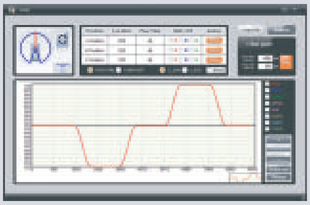
Developed with the ease of use in mind, Herkulex Smart Servo Series are the best 'Green' energy efficient smart servos in the world with low voltage, high output, asynchronous bi-directional communication, and more than 50 set-up parameters.

HerkuleX

ROBOGAMES
Apr 20~22, Visit our booth in Main Entrance Hall

Specification (DRS-0101, DRS-0201)

- **Dimensions** : 44.5mm(W) X 24.0mm(D) X 32.0mm(H) [1.75in. X 0.94in. X 1.26in.]
- **Weight** : 45g (DRS-0101) / 60g (DRS-0201) [1.59oz (DRS-0101) / 2.12oz (DRS-0201)]
- **Input Voltage** : 7.4V DC (DRS-0101) / 7~12V DC (Optimized 7.4V) (DRS-0201)
- **Stall Torque** : 12kgf.cm@7.4V (DRS-0101) / 24kgf.cm@7.4V (DRS-0201) [166.8 ozf.in. (DRS-0101) / 333.6 ozf.in. (DRS-0201)]
- **Maximum Speed** : 0.166s/60˚@7.4V (DRS-0101) / 0.147s/60˚@7.4V (DRS-0201)
- **Operating Angle** : 320˚, Continuous Rotation
- **Communication** : Full Duplex Asynchronous Serial(TTL), Multi Drop, 0-254 ID, Maximum Baud Rate : 0.67Mbps
- **Motor** : Metal Brush DC Cored (DRS-0101) / Coreless DC (DRS-0201)
- **Gear** : Super Engineering Plastic (DRS-0101) / Reinforced Metal (DRS-0201)
- **Feedback** : Position, Speed, Temperature, Load, Voltage, etc.
- **Features** : PID, Feedforward, Trapezoidal Velocity Profile, Velocity Override, Torque Saturator & Offset, Overload Protection, Neutral Calibration, Deadband, 54 Selectable Setting Parameters (Sold Separately : HerkuleX Manager Kit)

HerkuleX Manager is a bundled software that uses the GUI to maximize the ease of operation in setting up more than 50 servo operating parameters and servo maintenance using such a tool as the real time trend graph.

HOVIS Lite HOVIS Eco

Dongbu Robot

Dongbu Robot Co., Ltd. www.dongburobot.com

Headquarter : 11th Floor, Bucheon Techno Park 401, Yakdae-dong, Wonmi-gu, Bucheon-city, Gyunggi-do 420-734, Korea
TEL : +82-32-329-5551(ext.311), FAX : +82-32-329-5569, E-MAIL : robotsales@dongburobot.com
Factory : 27, 6 Gil, 4 Sandan, Jiksan-eup, Seobuk-gu, Cheonan-city, Chungcheongnam-do 331-814, Korea, TEL : +82-41-590-1700, FAX : +82-41-590-1701